



中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生数学建模竞赛

学　　校	河海大学
参赛队号	20102940042
队员姓名	1. 李　禹
	2. 徐文强
	3. 崔　威

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

题 目 **汽油辛烷值优化建模**

摘 要：

汽油燃烧产生的尾气排放对大气环境有重要影响。汽油清洁化重点是降低汽油中的硫、烯烃含量，同时尽量保持其辛烷值。然而汽油精制工艺是一个多变量的强非线性、强耦联性的复杂系统。为控制精制过程中产品的硫含量、建立汽油辛烷值预测模型和优化操作变量降低辛烷值(RON)损失，本文充分利用线性、非线性相关分析、互信息理论、神经网络、智能寻优算法等数据挖掘技术研究了如何降低汽油精制过程中辛烷值损失的问题。

针对问题一，依据附件二给出的“样本确定方法”对附件三：285 号和 313 号样本原始数据进行数据整定。经整定，两个样本原始数据中不存在残缺数据较多、全部为空值或部分为空值的位点（操作变量），不需作数据处理；依据变量的最大最小限幅，筛选出 8 个变量，剔除 4 个涉及样本数超过 42.15%的变量；将整定好的样本替换附件一中的对应样本，同时利用拉依达准则剔除 4 个存在非操作变量异常值的样本。最终形成 321 个样本、364 个变量的数据文件。

针对问题二，汽油精制工艺是一个多变量的强非线性、强耦联性的复杂系统，首先模型一利用简单线性相关分析和偏相关分析，得到产品辛烷值(RON)和原料辛烷值(RON)的强相关性，相关系数为 0.973。然后建立基于互信息的降维模型，将剩余变量依据与产品辛烷值(RON)互信息值的大小排序，剔除线性相关较强的变量后，选取前 14 个变量和原料辛烷值(RON)共计 15 个主要变量。模型二基于遗传算法—BP 神经网络的降维模型利用了神经网络对非线性问题的良好适应性，采用二进制编码形成自变量组合，通过 BP 神经网络计算得到产品辛烷值(RON)预测值的均方误差作为该自变量组合的适应度，经遗传算法寻优得到最优的 14 个主要变量的组合。对比两个模型，发现两组主要变量有 8 个完全相同，但模型一的剩余主要变量之间存在更弱的相关性，符合主要变量的独立性标准。

针对问题三，基于问题二模型一所得到的 15 个主要变量，建立模型三 BP 神经网络模型预测产品辛烷值(RON)。单隐含层 BP 神经网络预测误差在（-0.36%，0.55%）之间，双隐含层 BP 神经网络预测误差在（-0.4%，0.35%）之间。模型四小波神经网络预测模型的预测误差在（-0.15%，0.3%）。对比得知，小波神经网络的预测精度最高，但对比产品辛烷值预测值与实测值的整体趋势，发现双隐含层 BP 神经网络的预测误差分布更均匀，精度与小波神经网络预测模型接近。因此选定双隐含层 BP 神经网络作为预测模型。

针对问题四，基于问题三的产品辛烷值(RON)预测模型，同样建立产品硫含量的 BP 神经网络预测模型。建立模型五基于 BP 神经网络预测模型的粒子群寻优算法，优化主要变量操作方案。优化过程中，除原料辛烷值(RON)的速度为 0 以外，粒子中的其他主要变量的速度以 50%的概率取相应的单步调整幅度 $+\Delta_{x_i}$ 或 $-\Delta_{x_i}$，最大迭代次数 *MaxGen* 取为产品辛烷值(RON)测量前两个小时内，以 1 次/3min 的调整频次所允许的最大调整次数 40次 。假定合理的产品辛烷值目标区间为（0.5,0.7），最终优化得到全部 325 个样本的最优操作条件。

针对问题五，利用问题四建立的模型五，根据优化过程中存储的粒子位置矩阵和速度

矩阵，倒推得到操作变量最优解的变化轨迹，再利用问题三的 BP 神经网络预测模型得到过程中每个操作变量组合的产品辛烷值(RON)和硫含量的变化轨迹，并给出具体的操作变量轨迹。结果表明：133 号样本的产品辛烷值(RON)由初始的 88.09 个单位提高到 88.71 个单位，辛烷值损失降幅达到 52.67%，同时，调整过程中硫含量始终不超过$5.00\mu g/g$。

关键词：互信息理论、非线性相关分析、辛烷值预测、BP 神经网络、粒子群寻优算法

目录

一、 问题重述

汽油是小型车辆的主要燃料，汽油燃烧产生的尾气排放对大气环境有重要影响。汽油清洁化重点是降低汽油中的硫、烯烃含量，同时尽量保持其辛烷值。为进一步提高汽油产品的品质，依据采集的数据样本通过数据挖掘技术建立辛烷值(RON)损失预测模型，给出每个样本的优化操作条件，在保证汽油产品脱硫效果的前提下，使得汽油辛烷值(RON)损失降幅在 30%以上。

问题一：参考近 4 年的工业数据的预处理结果，依“样本确定方法”对 285 号和 313 号数据样本进行预处理并将处理后的数据分别加入到附件一相应的样本号中，供后续研究使用。

问题二：由于催化裂化汽油精制过程是连续的，虽然操作变量每 3 分钟就采样一次，但辛烷值（因变量）的测量比较麻烦，一周仅 2 次无法对应。但根据实际情况认为辛烷值的测量值是测量时刻前两小时内操作变量的综合效果，因此预处理中取操作变量两小时内的平均值与辛烷值的测量值对应，产生了 325 个样本。

建立降低辛烷值(RON)损失模型涉及包括 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、2 个产品性质等变量以及另外 354 个操作变量（共计 367 个变量），采用先降维后建模的方法，有利于忽略次要因素，发现并分析影响模型的主要变量与因素。因此，根据提供的 325 个样本数据，通过降维的方法从 367 个操作变量中合理筛选出建模主要变量，使之尽可能具有代表性、独立性。

问题三：建立辛烷值(RON)损失预测模型。采用上述样本和建模主要变量，通过数据挖掘技术建立辛烷值(RON)损失预测模型，并进行模型验证。

问题四：主要变量操作方案的优化。要求在保证产品硫含量不大于$5.00\mu g/g$的前提下，利用上述模型获得 325 个数据样本中，辛烷值(RON)损失降幅大于 30%的样本对应的主要变量优化后的操作条件（优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，以它们在样本中的数据为准）。

问题五：模型的可视化展示：工业装置为了平稳生产，优化后的主要操作变量（即问题二中的主要变量）往往只能逐步调整到位，对 133 号样本（原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，以样本中的数据为准），以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值(RON)和硫含量的变化轨迹。

二、 问题分析

问题一要求对附件三的数据按照“样本确定方法”进行处理，使得处理后的数据符合附件一的形式。本问将根据“样本确定方法”中的数据整定方法对 285 号和 313 号数据样本进行整定，并放入附件一对应的样本号中。同时对附件一中极个别异常值所在的样本应用拉依达准则予以剔除，形成便于数据挖掘的样本数据。

问题二要求对模型涉及的 364 个变量进行降维，找到与降低产品辛烷值(RON)损失最相关的多个变量，并说明选择变量的过程及合理性。问题二中的样本变量包括原料性质、待生吸附剂性质、再生吸附剂性质、产品性质以及操作变量，考虑模型的目标结果应为产品性质的两个变量，即产品辛烷值(RON)和硫含量。如果将产品性质与其他变量当作一个函数映射关系，那么将两者作为模型因变量，其余 364 个变量作为自变量，并对因变量和自变量作相关性分析，选择相关系数较大的变量作为主要变量。由于炼油工艺过程的复杂性等原因，其操作变量与因变量是高度非线性的关系，而原料性质等其余自变量与因变量可能是线性关系。可以采用相关性分析，选择与产品辛烷值(RON)和硫含量线性相关度较高的变量；采用基于互信息的非线性相关分析，选择与产品辛烷值(RON)和硫含量非线性相关度较高的变量，并采用遗传算法进行优化选择，从而得到建模的主要变量。

问题三要求建立辛烷值(RON)损失预测模型。采用整定后的 321 个样本和建模主要变量，通过数据挖掘技术建立辛烷值(RON)损失预测模型，并进行模型验证。本问建立 BP 神经网络模型预测产品辛烷值(RON)。考虑单隐含层 BP 神经网络、双隐含层 BP 神经网络以及小波神经网络分别进行预测并予以比较。

问题四要求对主要变量的操作方案进行优化。在保证产品硫含量不大于 $5.00\,\mu g/g$ 的前提下，优化得到辛烷值(RON)损失降幅大于 30%的样本对应的主要变量优化后的操作条件。利用产品辛烷值(RON)预测模型和产品硫含量的 BP 神经网络预测模型，作为基于 BP 神经网络预测模型的粒子群寻优算法的适应度函数，优化主要变量操作方案。

问题五要求对优化模型进行可视化展示，以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值(RON)和硫含量的变化轨迹。利用问题四建立的模型，根据优化过程中存储的粒子位置矩阵和速度矩阵，倒推得到操作变量最优解的变化轨迹，再利用问题三的 BP 神经网络预测模型得到过程中每个操作变量组合的产品辛烷值(RON)和硫含量的变化轨迹，并给出具体的操作变量轨迹。

三、 模型假设

1、 假设测量数据都是真实数据，存在各种可能的误差。

2、 假设操作变量的取值范围完全参照附件四。

3、 假设任意操作变量 x_i 一次调整的幅度只能是 $\pm\Delta_{x_i}$ 。

4、 假设任意操作变量 x_i 每次正向调整或负向调整方向的概率是 50%。

5、 假设操作变量按照与采集频率相同的频率进行调整，即 1 次/3min。

四、 模型建立与求解

4.1 问题一：样本数据处理

4.1.1 思路概述

原始数据是从实际操作过程中直接测量得到，由于装置和测量方法等存在误差、部分位点存在问题，因此需要依据题目要求对样本数据进行处理。问题一要求参考附件一的预处理结果，以附件二所提到的“样本确定方法”中所提及的 5 点要求，对附件三中的 2 个原始样本数据进行筛选处理，并替代附件一中相应样本。

4.1.2 数据处理

首先对于附件三中的数据按照附件二“样本确定方法”进行处理，将附件三中 285 号的操作变量以及 313 号的操作变量读入 MATLAB 中，生成 40 行 354 列的矩阵，然后对矩阵进行操作。

第一个要求是对于残缺数据较多的位点进行删除处理。经程序搜索测量得到的数据，并无残缺数据，虽然有部分零值，但这些零值在操作范围之内，因此是合理的。

第二个要求是删除 325 个样本中数据全部为空值的位点。将附件一中的数据全部读入 MATLAB 中生成 325 行 368 列的数据，经程序搜索，不存在空值的情况。因此，不需要对附件一中 325 个样本数据作删除位点的处理。

第三个要求是对于部分数据为空值的位点，空值处用其前后两个小时数据的平均值代替。实际上，由于 285 号以及 313 号样本数据无空值，因此不需要删除或使用前后两个小时数据的平均值进行替代。

第四个要求是根据工艺要求与操作经验，总结出原始数据变量的操作范围，然后采用最大最小的限幅方法剔除一部分不在此范围的操作变量。首先依据附件四的 354 个操作变量取值范围得到每个操作变量的最大、最小值，对于超过限幅的数据所对应的操作变量进行剔除。将第二个要求建立的 325×368 的矩阵编入 MATLAB 代码中，按其顺序从 1~368 依次进行列编号，设定最大、最小值作为筛选的范围，将超出筛选范围的样本数据挑选出来，挑选得到的异常数据见表 4.1。

表 4.1 最大、最小值检验异常的样本

异常变量代号	S-ZORB.AT_5201.PV	S-ZORB.SIS_LT_1001.PV	S-ZORB.AI_2903.PV	S-ZORB.FT_1202.TOTAL
异常变量标号	36	63	98	120
异常样本数	147	325	315	10
异常变量代号	S-ZORB.FT_1204.TOTAL	S-ZORB.TE_2001.DACA	S-ZORB.AT-0012.DACA.PV	S-ZORB.CAL.LEVEL.PV
异常变量标号	125	252	337	357
异常样本数	137	1	9	1

从中可以看出，第 36 列（S-ZORB.AT_5201.PV）、第 63 列（S-ZORB.SIS_LT_1001.PV）、第 98 列（S-ZORB.AI_2903.PV）和第 125 列（S-ZORB.FT_1204.TOTAL）的异常数据所对应的样本数量较多，应当予以删除；而第 120 列（S-ZORB.FT_1202.TOTAL）、第 252 列（S-ZORB.TE_2001.DACA）、第 337 列（S-ZORB.AT-0012.DACA.PV）和第 357 列（S-ZORB.CAL.LEVEL.PV）的异常数据所对应的样本数量均较少，对整体样本数影响不大，可能属于部分测量误差，因此可以保留。在将有问题的操作变量剔除后，得到处理后的数组为 364 列。其具体组成是 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、3 个产品性质和 350 个操作变量，一共组成 364 列数组。值得注意的是，附件一的数据原本在 285、313 行已经存在数据，用整定完成的附件三的样本数据替代。

第五个要求是对于附件一的非操作变量中的原料及产品辛烷值(RON)、辛烷值(RON)损失这三列数据，采用拉依达准则剔除异常的样本数据。按照贝塞尔公式(1)计算这 3 列数据每一列的标准差 σ，将剩余误差的绝对值超过 3σ 的样本剔除。采用上述方法发现其中的 14、142、179、185 行的数据存在异常，其异常值见表 4.2。

$$\sigma=[\frac{1}{n-1}\sum_{i=1}^{n}v_i^2]^{1/2}=\{[\sum_{i=1}^{n}x_i{}^2-(\sum_{i=1}^{n}x_i)^2/n]/(n-1)\}^{1/2} \quad (1)$$

表 4.2 3σ 原则筛选得到的异常值

样本行数	原料辛烷值		产品辛烷值(RON)		产品辛烷值(RON)损失	
	剩余误差绝对值	3σ 值	剩余误差绝对值	3σ 值	剩余误差绝对值	3σ 值
14	2.9015	2.8502	3.0468	2.9542	0.1452	0.6777
142	4.4015		3.3468		1.0548	
179	0.3985		1.1132		0.7148	
185	2.9015		3.0468		0.1452	

可以看出这 4 个样本均存在不满足 3σ 原则的变量值，因此需要进行剔除处理。从而得到最终处理好的附件一数据，为 321 个样本，一共 364 列变量数据，生成一个 321 行×364 列矩阵，之后的数据挖掘均是基于此数据样本文件进行处理。

表 4.3 数据筛选结果

序号	数据筛选要求	数据筛选结果
1	删除残缺数据较多点位	无残缺数据点，没有删除数据
2	删除 325 个样本中数据为空值的位点	无空值情况，无需对样本数据处理
3	去除附件三中数据为空值的位点，并用平均值代替	无空值情况，无需对样本数据处理
4	根据最大、最小范围去除附件一变量异常值	去除第 36 列（S-ZORB.AT_5201.PV）、第 63 列（S-ZORB.SIS_LT_1001.PV）、第 98 列（S-ZORB.AI_2903.PV）和第 125 列（S-ZORB.FT_1204.TOTAL）的数据
5	根据 3σ 原则筛选样本非操作变量异常值	去除第 14、142、179 和 182 行样本数据

4.2 问题二：寻找建模主要变量

4.2.1 思路概述

问题二要求根据处理得到的 321 个样本、364 个变量的信息，寻找具有代表性、独立性的主要变量，使得选择的主要变量可以较好地刻画产品辛烷值(RON)以及辛烷值(RON)损失的变化。由于选择的变量需要具有代表性，因此需要选择与辛烷值(RON)相关性强的变量，选择变量具有独立性则要求选择的变量之间的相关性较弱。

由于汽油炼制工艺过程复杂性以及设备的多样性，本质上汽油精制过程是一个强非线性、多变量耦联的复杂系统，由于变量数量较多，如果不对各类输入的变量进行筛选，将会导致“维数灾难”，从而使得模型精度降低，建模时间较长，甚至模型算法失效。因此，我们从汽油精制过程中各变量优化选取的角度采用两种方法分别进行变量降维，在大量原始建模数据中，对不同变量组合进行筛选，从而得到影响产品辛烷值(RON)和辛烷值(RON)损失的主要变量。

4.2.2 数据处理

本问题解答将基于问题一处理后的样本数据，考虑该样本为多维非线性，因此采用不同降维方法对其进行计算分析，对比降维结果，选择合适的主要变量。为方便建模计算，将所有变量数据进行编码，在附件一的样本中将变量从 1~364 顺序编号，即表 4.4 所示。

表 4.4 样本变量编码

变量名	原料硫含量	原料辛烷值	饱和烃	烯烃	芳烃	溴值	密度	…	8.0MPa 氢气至反吹氢压缩机出口	D101 原料缓冲罐压力
编码值	1	2	3	4	5	6	7	…	363	364

在此基础上，采用问题一得到的 321×364 样本矩阵进行下面模型的建立求解。

4.2.3 模型一：基于互信息的非线性相关分析模型

在炼油工艺过程中，原料性质、待生吸附剂和再生吸附剂性质和操作变量等与产品辛烷值(RON)可能存在线性相关性。因此，将样本数据标准化后，先对其进行线性相关分析，选择线性相关性较强的变量作为一部分主要变量。然后对剩余变量进行基于互信息的非线性相关分析，选取互信息值较大的变量作为主要变量。

（1）线性相关分析

对于多维样本线性相关性研究常用的方法有简单相关分析和偏相关分析。简单相关分析可以计算两个变量间的线性相关性，偏相关分析则是对多变量的影响进行控制和消除，从而分析两个变量间的净相关关系。

简单相关分析一般是通过计算因素之间的相关系数来分析两者的关系，可采用皮尔逊相关系数分别计算产品辛烷值(RON)与各个变量的相关性，皮尔逊相关系数的计算公式为：

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2(y_i-\bar{y})^2}} \tag{2}$$

式中：n 为样本量；x_i、y_i 是两个变量的观测值；$\bar{x}$、$\bar{y}$ 是变量的均值；r 是两个变量间的相关系数，其绝对值越接近 1 则变量间的相关性越大，正号为正相关，负号为负相关。

偏相关分析是在控制其他变量线性影响的条件下分析两个变量间的线性相关性，以三个变量为例，在控制其中一个变量的作用后采用一阶偏相关系数分析另两个变量间的相关性：

$$r_{ij\bullet h}=\frac{r_{ij}-r_{ih}r_{jh}}{\sqrt{(1-r_{ih}^2)(1-r_{jh}^2)}} \tag{3}$$

式中：r_{ij} 为变量 x_i 和 x_j 的简单相关系数，r_{ih} 为变量 x_i 和 x_h 的简单相关系数，r_{jh} 为变量 x_j 和 x_h 的简单相关系数。

在此基础上推广，对于四个变量情况，应通过控制两个变量的影响计算剩余两个变量的偏相关系数，即：

$$r_{ij\bullet hm}=\frac{r_{ij\bullet h}-r_{im\bullet h}r_{jm\bullet h}}{\sqrt{(1-r_{im\bullet h}^2)(1-r_{jm\bullet h}^2)}} \tag{4}$$

二阶偏相关系数是在一阶偏相关系数的基础上建立得到，本问题对产品辛烷值(RON)与变量作偏相关分析，并进行显著性检验。

显著性检验是反映样本统计量和假设总体参数间的显著性差异，通过小概率原理先假定非线性，对样本进行检验。一般地，当计算所得的显著性概率小于 α=0.05 时，即代表两个变量间存在显著的线性相关性，反之没有明显线性相关性。

使用 SPSS 软件和 MATLAB 软件对 361 个自变量(7 个原料性质、4 个吸附剂性质以及 350 个操作变量)和 2 个因变量(产品辛烷值(RON)、硫含量)做标准化处理并进行简单相关

分析，计算与产品中含硫量和辛烷值的相关系数，其部分代表性结果如下所示：

表 4.5 产品性质的简单相关分析

	产品硫含量皮尔逊相关性	产品辛烷值皮尔逊相关性
原料硫含量	0.233	0.495
原料辛烷值	0.158	0.973
饱和烃	-0.327	-0.465
烯烃	0.370	0.395
芳烃	-0.180	0.093
溴值	0.081	-0.129
密度	-0.004	0.261
…	…	…
8.0MPa 氢气至反吹氢压缩机出口	-0.205	-0.180
D101 原料缓冲罐压力	-0.399	-0.214

可明显看出除了产品辛烷值(RON)与原料辛烷值(RON)存在明显线性相关性，其余变量与 2 个因变量之间无明显线性相关。

考虑到变量间可能存在线性相关，如果不作偏相关分析，可能会造成信息的重复表达、主要变量过多等问题。因此对原料性质、待生吸附剂和再生吸附剂性质等 11 个变量做偏相关分析，其结果见表 4. 6：

表 4.6 产品性质偏相关分析

	产品硫含量		产品辛烷值(RON)	
	偏相关性	显著性	偏相关性	显著性
原料硫含量	0.105	0.064	0.112	0.048
原料辛烷值(RON)	0.007	0.899	0.965	0.000
饱和烃	0.029	0.613	-0.070	0.216
烯烃	0.031	0.585	-0.069	0.225
芳烃	0.027	0.636	-0.068	0.227
溴值	0.075	0.182	-0.046	0.420
密度	0.038	0.498	0.044	0.442
待生吸附剂焦炭	0.091	0.109	0.151	0.007
待生吸附剂 S	-0.035	0.534	-0.102	0.072
再生吸附剂焦炭	-0.065	0.253	-0.220	0.000
再生吸附剂 S	0.058	0.307	0.176	0.002

偏相关分析计算结果表明产品辛烷值(RON)与原料辛烷值(RON)存在明显线性相关性。偏相关分析剔除了自变量之间的线性相关性，使得筛选出的变量独立性更高，更具代表性。在简单相关性分析中，产品辛烷值与原料辛烷值的线性相关系数为 0.973，偏相关分析中，线性相关系数为 0.965，且显著性系数为 0，满足显著性检验，代表其具有显著代表性。因此可以将原料辛烷值(RON)作为主要变量之一。

将 321 个样本中的产品辛烷值(RON)与原料辛烷值(RON)绘制简单散点图（见图 4.1），可发现两者之间呈现明显线性关系，因此上述计算分析是准确的，而其余自变量与因变量的相关系数较低，以线性相关度第二高的硫含量为例（见图 4.2），其线性相关程度不明显，不能通过线性分析确定其是否属于主要变量。

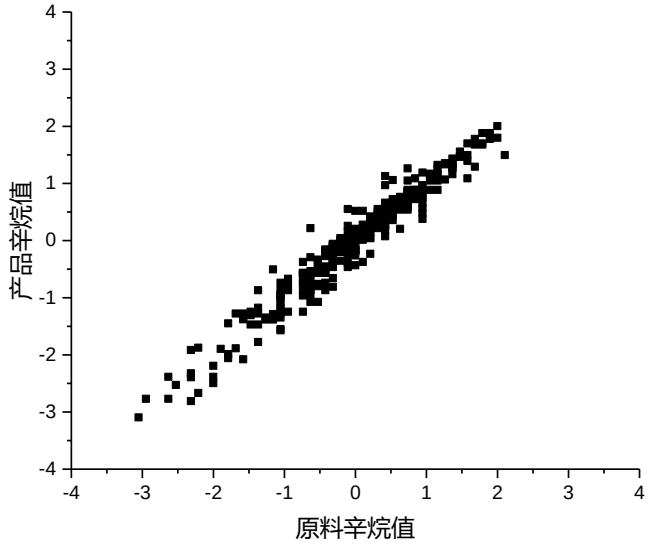

图 4.1 产品辛烷值与原料辛烷值线性分布图

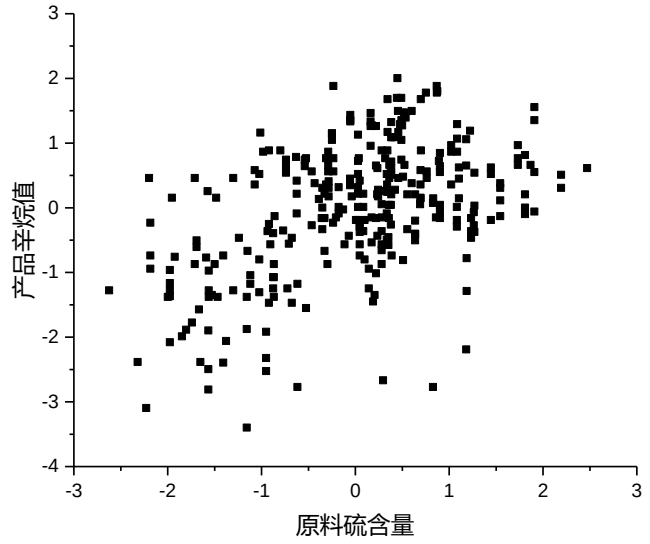

图 4.2 产品辛烷值与原料硫含量线性分布图

（2）基于互信息[1]的相关分析

汽油精制过程中的变量具有高度非线性和相互强耦联性，传统的线性降维方法包括主成分分析法等无法计算变量间的非线性关系。而互信息方法是用来描述变量间相互关联大小，既可以表现变量间的线性关系，又可以反映出变量间的非线性关系。当变量间的互信息值越大时，即代表其相互耦联性越强。

首先假定 X 为一个离散随机变量， $p(X)$ 表示变量 X 取值为 x 的概率，则其不确定性程度可以表示为信息熵 $H(X)$：

$$H(X)=-\sum_{x\in X}p(x)\log p(x) \tag{5}$$

式中： x 是变量 X 的可能取值。当 x 的分布高度偏向某个特定值时，即 x 的取值不确定性小，熵值较低；若其分布均匀则代表其不确定性较大，熵值较大。

在随机变量 X 给定的条件下随机变量 Y 的条件熵为：

$$H(Y|X) = -\sum_{y\in Y} p(y) \sum_{x\in X} p(x|y) \log p(x|y) \tag{6}$$

式中： $p(x|y)$ 为 Y 已知情况下 X 的条件概率分布。

互信息量化了一个随机变量可以通过另一个随机变量获得的信息量，因此对于 X 和 Y 之间的互信息为：

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{7}$$

式中： $p(x,y)$ 为联合概率密度函数， $p(x)$ 和 $p(y)$ 分别为对应的边际概率密度函数。关于互信息的结构见图 4.3。

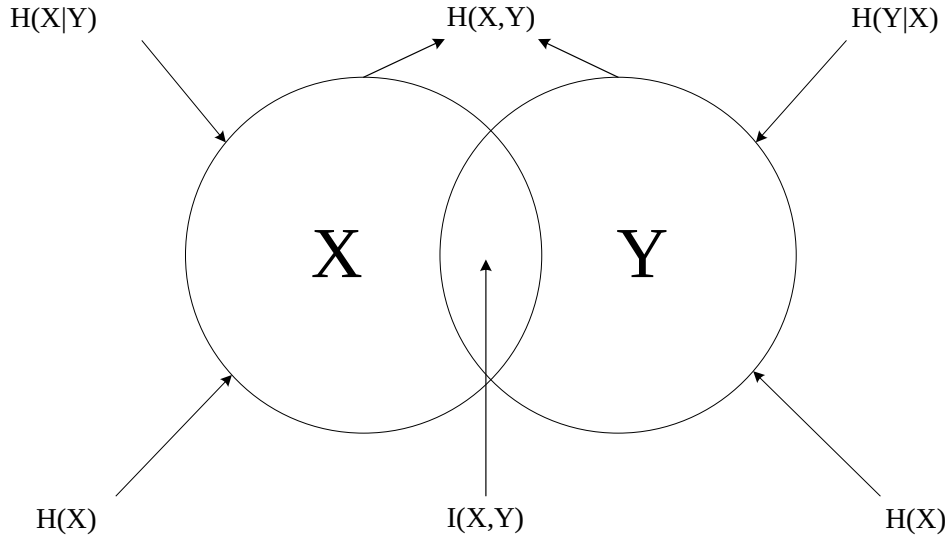

图 4.3 互信息示意图

依据问题一处理得到的 321×364 样本矩阵。采用 MATLAB 软件计算因变量与每个变量的互信息值，并进行从大到小排序，计算结果见表 4.7。

表 4.7 因变量（3 个都给出）与自变量互信息值

变量编码号	变量互信息值		变量编码号	变量互信息值
363	0.865		309	0.288
105	0.865		310	0.265
119	0.864		218	0.246
102	0.863	…	100	0.237
108	0.860		168	0.217
361	0.856		226	0.206
111	0.856		215	0.194
125	0.853		204	0.181
104	0.852		177	0.177

根据表 4.7 结果可以选择与产品辛烷值(RON)非线性相关度较高的自变量，理论上互信息值较大的变量对于因变量的影响是较大的，是具有代表性的变量。但实际上，这些变量间相关性较高，即这些变量代表的可能是同一性质的操作，这一类性质的改变对于产品辛烷值的影响较大，所以它们的互信息量都处于较大的范围。为了使得选择的主要变量尽可能的具有代表性和独立性，所以需要在互信息分析的基础上去除掉自变量之间相关性高的部分，从而保证选择的变量不仅对因变量的相关度较高，而且各变量之间尽可能不相关，能够较为全面地代表操作变量整体。

在对自变量进行互信息值计算排序后，依次计算变量间的相关系数，从第一个变量开

始，计算其与第二个变量的相关系数，若相关系数>0.50，则代表这两个变量是中度相关的，其不具备充分的独立性，所以需要去掉互信息值较低的第二个变量，继续计算第一个变量与第三个变量的相关系数，即考虑是否选择该变量作为主要变量的时候需要计算与已经选择的主要变量的全部相关性，使得新添加的主要变量与前面所有主要变量的相关性较低。如此保证选择主要变量的独立性。自变量之间的相关性见表 4.8。

表 4.8 各个自变量之间的相关系数

变量编码	363	105	119	102	108	361	…	177
363	1.0000	-0.0387	-0.0363	-0.0262	-0.0593	0.9947	…	0.0591
105	-0.0387	1.0000	0.9989	0.9990	0.9995	-0.0432	…	-0.0470
119	-0.0363	0.9989	1.0000	0.9999	0.9991	-0.0392	…	-0.0482
102	-0.0262	0.9990	0.9999	1.0000	0.9988	-0.0289	…	-0.0480
108	-0.0593	0.9995	0.9991	0.9988	1.0000	-0.0644	…	-0.0483
361	0.9947	-0.0432	-0.0392	-0.0289	-0.0644	1.0000	…	0.0567
111	-0.1210	0.9958	0.9963	0.9953	0.9976	-0.1237	…	-0.0528
…	…	…	…	…	…	…	…	…
204	-0.2222	0.0755	0.0869	0.0866	0.0823	-0.2009	…	0.0071
177	0.0591	-0.0470	-0.0482	-0.0480	-0.0483	0.0567	…	1.0000

按照互信息值大小排序后的变量顺序依次进行对比挑选，最终挑选剩下的自变量为 91 个变量，其挑选结果见表 4.9。

表 4.9 去除自相关度的变量互信息值

变量编码号	变量互信息值		变量编码号	变量互信息值
363	0.865		138	0.436
105	0.865		142	0.397
271	0.831		189	0.388
214	0.825		222	0.362
133	0.824	…	218	0.246
34	0.820		168	0.217
40	0.818		215	0.194
73	0.816		204	0.181
83	0.816		177	0.177

在这 91 个自变量里，根据互信息值大小，选择指定数量的变量作为主要变量。综合线性分析和非线性分析结果，选择线性相关度和非线性相关度较高的变量作为主要变量，其主要变量见表 4.10。

表 4.10 主要变量选择

变量编码值	变量代号	变量名称
2		辛烷值 RON
363	S-ZORB.FT_1504.TOTALIZERA.PV	8.0MPa 氢气至反吹氢压缩机出口
105	S-ZORB.FT_5101.TOTAL	-
271	S-ZORB.TE_7506B.DACA	K-103B 进气温度
214	S-ZORB.PT_5201.DACA	精制汽油出装置线压力
133	S-ZORB.LI_9102.DACA	D-204 液位
34	S-ZORB.TE_5202.PV	精制汽油出装置温度
40	S-ZORB.FT_5101.PV	干气出装置流量
73	S-ZORB.TE_1608.PV	加热炉循环氢出口温度
83	S-ZORB.TE_1203.PV	D121 温度
262	S-ZORB.PDT_3002.DACA	ME-105 过滤器压差
297	S-ZORB.PT_6005.DACA	F-101 辐射室底部压力
56	S-ZORB.PDC_2502.PV	D107 转剂线压差
145	S-ZORB.FT_3303.DACA	D-123 蒸汽出口流量
205	S-ZORB.FC_5103.DACA	稳定塔顶回流流量

对选择的主要变量组成进行分析，其与产品辛烷值(RON)不仅具有较高的相关度，而且主要变量之间的线性相关度也较低，因此选择出来的主要变量具有代表性和独立性，可以满足建模需求。

4.2.4 模型二：基于遗传算法-BP 神经网络的降维模型[2]

模型一充分利用了互信息理论和相关性分析的理论，选择与产品辛烷值(RON)相关性较大的变量，但主要变量的数量由人为指定，难以保证其客观性，存在一定程度的主观因素。因此，引入基于遗传算法-BP 神经网络的降维模型对变量降维方法进行优化，题目要求主要变量在 30 个以内，为减少模型计算量，我们先计算因变量与各自变量的线性相关度和非线性相关度，筛选出线性相关度较高的 3 个变量和 25 个非线性相关度较高的变量。将样本中其他的变量删除，保留筛选得到的 28 个变量，并对样本做标准化处理，其对应变量的计算公式为：

$$\overline{x_i} = \frac{x_i - \mu}{\delta} \tag{8}$$

式中：$\overline{x_i}$ 为标准化后第 i 个样本值；x_i 为第 i 个样本；μ 为该变量的样本均值；δ 为该变量的样本标准差。

模型二利用遗传算法可以模拟自然选择和交叉变异等现象的特点，构建不同的自变量组合并不断迭代优化，最终选择拟合效果好的自变量组合参与建模。模型的设计步骤如图 4.3 所示。

基于遗传算法-BP 神经网络的降维模型的主要步骤如下：

（1）初始种群的产生。随机生成 50 个 28 位的二进制数，每个二进制数不同数位上的 1 或者 0 代表是否将对应该位置的变量选为主要变量，50 为种群的个体数量。28 个变量的顺序编码如表 4.11 所示。

表 4.11 28 个变量的顺序编码对应

顺序	1	2	3	4	5	……	27	28
变量编码	2	58	1	363	105	……	299	12

（2）适应度函数。遗传算法中使用适应度来度量种群中各个个体在优化计算中可能达到、接近或有助于找到最优解的优良程度。适应度较高的个体遗传到下一代的概率就会相对较大。适应度函数是适应度的度量。这里采用数据集误差平方和的倒数作为每个个体的适应度 f_i，即：

$$f_i = \frac{1}{\sum_{i=1}^{n}(\overline{y_i} - y_i)^2} \tag{9}$$

式中：$\overline{y_i}$ 是第 i 个样本预测值；y_i 是第 i 个样本真实值；n 为样本数目。

特别地，样本预测值 $\overline{y_i}$ 通过当前个体中自变量的数量建立相应结构的 BP 神经网络来求解。为避免初始权值和阈值的随机性对适应度函数计算的影响，针对每一个个体计算适应度函数值时，均用遗传算法对所建立的 BP 神经网络的权值和阈值进行优化，优化步骤如图 4.3 所示。

BP 神经网络是一种神经网络学习算法。将遗传算法与 BP 神经网络相结合的目的在于将对应当前个体的动态结构 BP 神经网络作为一个函数，计算遗传算法的适应度，从而将预测效果显著的自变量带入神经网络模型中。我们通过三层网络结构进行变量筛选，即输入层到隐含层用 Tansig 为训练函数，隐含层到输出层以 purelin 为传递函数，采用非线性的 Levenberg-Marquardt 算法作为训练算法。

在本模型中，将每个个体代入 BP 网络中计算，得到一个对应的均方误差，将其倒数作为适应度函数代入遗传算法进行选择计算。

（3）选择运算

从当前的种群里选择适应度较高的个体进行下一步的繁衍，因此适应度高的个体被选中的概率就越大。我们采用轮盘赌法，即基于适应度比例的选择策略。每个个体被选中的概率为：

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{10}$$

式中：N 为种群中个体总数目。

为了提高遗传算法的收敛性，采用最优保留策略，将最大适应度的个体直接保留到下一代。每次更新种群时将群体中最差的个体进行替换成上一代的最优个体，以防止当前种群中适应度较好的个体被淘汰。

（5）交叉操作

对于自变量的压缩降维，采用最简单的单点交叉算子。单点交叉算子原理如图 4.4。

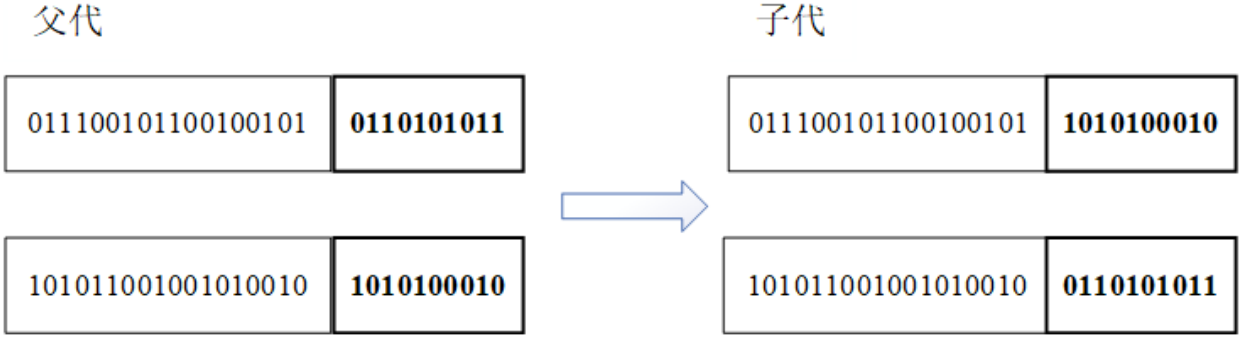

图 4.4 单点交叉算子示意图

对 BP 神经网络的初始权值和阈值优化时采用算术交叉算子，先对种群中的个体进行两两随机配对，通过交叉概率产生两个新个体，即：

$$c_1 = p_1 \times a + p_2 \times (1-a) \tag{11}$$

$$c_2 = p_2 \times a + p_1 \times (1-a) \tag{12}$$

式中：c_1、c_2 为交叉后产生的子代；a 为（0，1）间的随机数即为交叉概率；p_1、p_2 为上一组配对的两个父代。

（6）变异操作

对于自变量的降维操作采用单点变异算子，通过随机产生变异点，改变对应基因座上的基因值，对于本案中即显示为：“0”变为“1”或者“1”变为“0”，如图 4.5 所示。通过不断循环更新个体，从而避免陷入局部最优。

图 4.5 单点变异算子操作示意图

（7）终止规则

采用最大迭代次数作为终止条件，最大迭代次数设为 100。

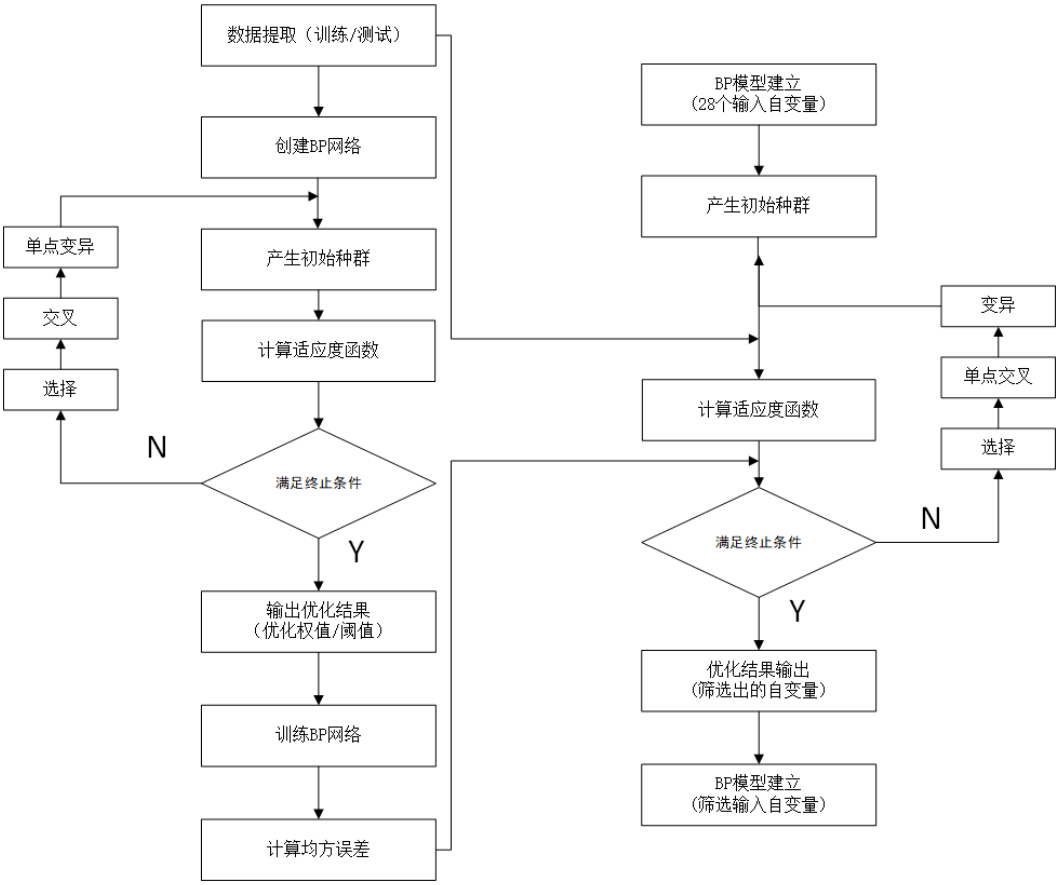

图 4.6 遗传算法-BP 神经网络流程图

通过基于遗传算法-BP 神经网络的降维模型对影响产品辛烷值(RON)的 28 个变量做降维处理，利用 MATLAB 程序进行迭代计算，得到的最优二进制编码见表 4.12。

表 4.12 最优二进制编码表

序号	1	2	3	4	5	6	7
最优二进制	1	0	0	1	1	1	1
变量编码	2	58	1	363	105	271	214
序号	8	9	10	11	12	13	14
最优二进制	1	0	0	1	0	1	0
变量编码	133	34	40	73	83	262	297
序号	15	16	17	18	19	20	21
最优二进制	0	0	0	1	0	1	1
变量编码	56	145	205	171	232	179	19

序号	22	23	24	25	26	27	28
最优二进制	0	1	0	0	0	1	1
变量编码	285	295	209	47	12	260	299

模型二的遗传算法优化降维过程如图 4.7 所示，由图可知，当种群进化到第 80 代之后得到了最优个体，种群的平均适应度函数稳定较好，部分的波动可能是算法中的一些随机性因素造成的。

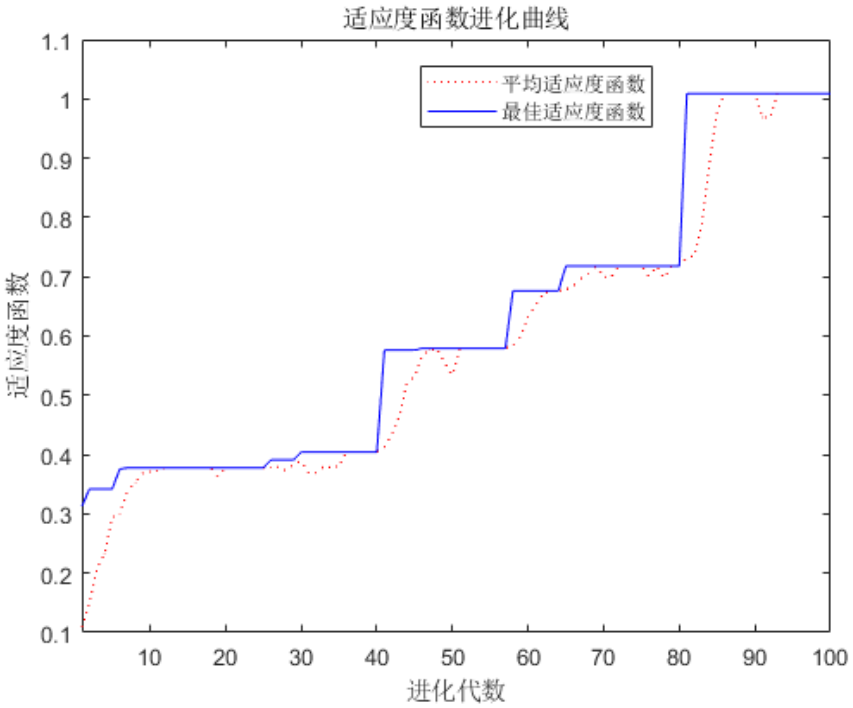

图 4.7 种群适应度函数进化曲线

4.2.5 主要变量对比

将模型一和模型二选择的主要变量进行对比见表 4.13。

表 4.13 主要变量对比

模型一			模型二		
变量编码值	变量代号	变量相关度	变量编码值	变量代号	变量相关度
2	-	0.973	2	-	0.973
363	S-ZORB.FT_1504.TOTALIZERA.PV	0.865	363	S-ZORB.FT_1504.TOTALIZERA.PV	0.865
105	S-ZORB.FT_5101.TOTAL	0.865	105	S-ZORB.FT_5101.TOTAL	0.865
271	S-ZORB.TE_7506B.DACA	0.831	271	S-ZORB.TE_7506B.DACA	0.831
214	S-ZORB.PT_5201.DACA	0.825	214	S-ZORB.PT_5201.DACA	0.825
133	S-ZORB.LI_9102.DACA	0.824	133	S-ZORB.LI_9102.DACA	0.824
34	S-ZORB.TE_5202.PV	0.820	73	S-ZORB.TE_1608.PV	0.820
40	S-ZORB.FT_5101.PV	0.818	262	S-ZORB.PDT_3002.DACA	0.818
73	S-ZORB.TE_1608.PV	0.816	171	S-ZORB.TE_2401.DACA	0.810
83	S-ZORB.TE_1203.PV	0.816	179	S-ZORB.TE_5004.DACA	0.803
262	S-ZORB.PDT_3002.DACA	0.815	19	S-ZORB.TE_2103.PV	0.802
297	S-ZORB.PT_6005.DACA	0.814	295	S-ZORB.PT_1604.DACA	0.801
56	S-ZORB.PDC_2502.PV	0.812	260	S-ZORB.PDT_3502.DACA	0.799
145	S-ZORB.FT_3303.DACA	0.812	299	S-ZORB.PT_1601.DACA	0.798
205	S-ZORB.FC_5103.DACA	0.810	-	-	

从上表对比结果我们可以看出，模型一选择了 15 个变量，模型二选择了 14 个变量，

两者对比有前 8 个变量是一样的，这说明变量选择方向是正确的。在最终变量的选择上，我们认为参照模型一的选择结果较为合理，原因是模型二虽然不受人为指定主要变量数目的限制，但存在一定的随机性，特别是挑选得到的后面几个主要变量与产品辛烷值(RON)的非线性相关度与模型一相比较小，因此其变量的代表性较模型一会显得不足。其次，模型二与模型一中的变量有一定的重叠度，且重叠部分都是前几个变量，这些变量与产品辛烷值(RON)的相关度较高，具有代表性，说明模型一选择的变量是正确的。而模型一同时也考虑到剔除变量自身的相关性，这保证了变量选择的独立性要求。因此，我们选择模型一的主要变量挑选结果作为下面建模的主要变量，能够满足题目中的变量选择要求，其主要变量选择结果见表 4.14。

表 4.14 主要变量选择结果

序号	变量编码值	变量代号	变量名称
1	2	-	辛烷值 RON
2	363	S-ZORB.FT_1504.TOTALIZERA.PV	8.0MPa 氢气至反吹氢压缩机出口
3	105	S-ZORB.FT_5101.TOTAL	-
4	271	S-ZORB.TE_7506B.DACA	K-103B 进气温度
5	214	S-ZORB.PT_5201.DACA	精制汽油出装置线压力
6	133	S-ZORB.LI_9102.DACA	D-204 液位
7	34	S-ZORB.TE_5202.PV	精制汽油出装置温度
8	40	S-ZORB.FT_5101.PV	干气出装置流量
9	73	S-ZORB.TE_1608.PV	加热炉循环氢出口温度
10	83	S-ZORB.TE_1203.PV	D121 温度
11	262	S-ZORB.PDT_3002.DACA	ME-105 过滤器压差
12	297	S-ZORB.PT_6005.DACA	F-101 辐射室底部压力
13	56	S-ZORB.PDC_2502.PV	D107 转剂线压差
14	145	S-ZORB.FT_3303.DACA	D-123 蒸汽出口流量
15	205	S-ZORB.FC_5103.DACA	稳定塔顶回流流量

4.3 问题三：产品辛烷值预测

问题三是对汽油精炼过程中产品辛烷值(RON)以及辛烷值(RON)损失进行预测，即在问题一和问题二的基础上，针对问题一挑选出来的样本，采用问题二筛选出来的主要变量，建立预测辛烷值的模型。此过程分为两部分：

其一：建立产品辛烷值(RON)预测模型[3]。该部分采用基于 BP 预测模型进行建模，计算主要变量与产品辛烷值(RON)的预测关系。

其二：对上一步建立的预测模型进行优化，优化的目标是使预测的产品辛烷值(RON)和实际产品辛烷值(RON)的差距尽可能小。主要方法包括：1、增加隐含层数，调整模型的节点数，节点数取值从 3~11。随机选取 300 个样本作为训练集、21 个样本作为检验集，重复 100 次选取过程，得到 100 次计算的平均计算误差，如表 4.15 所示。以提高模型精确度；2、考虑采用小波神经模型预测产品辛烷值(RON)，与前者进行对比。

表 4.15 隐含层节点数的平均计算误差

隐含层节点数	平均计算误差	隐含层节点数	平均计算误差
3	0.1608	8	0.1756
4	0.1632	9	0.1719
5	0.1614	10	0.1703
6	0.1667	11	0.1795
7	0.1679		

4.3.1 模型三：BP 神经网络预测模型

BP 神经网络是通过误差的反向传播算法，训练多层前馈网络，经过训练掌握学习样本

的输入输出映射关系。在一般的正向传播中，输入信号经过输入层、隐含层和输出层，而反向传播是由输出层开始逐层计算各层神经元的输出误差，采用误差梯度下降法在保证误差信号最小的前提下，来修改各层神经元的权重和阈值。调整 BP 神经网络的训练函数和传递函数，可以实现复杂的非线性映射问题。本文利用神经网络的这一特性，从而较为精准地预测产品辛烷值(RON)。

BP 网络各个拓扑结构之间用全连接方式实现，处于相同层的神经元相互之间处于断开状态，不同层之间用的全互连，一般情况下至少有一个隐含层，隐含层向下传递采用的是 Sigmoid 型函数，BP 网络拓扑结构见图 4.8。

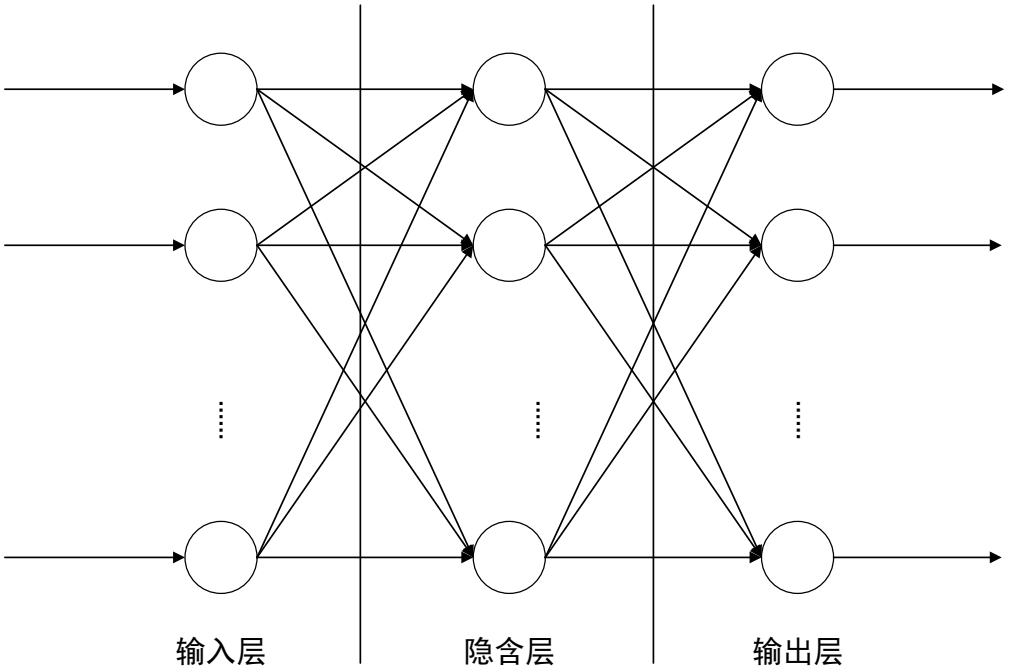

图 4.8 BP 网络拓补结构示意图

当样本从输入层经过隐含层再到输出层时，首先是是将样本值代入 Sigmoid 函数，即：

$$f(x)=\frac{1}{1+\exp[-(x+\theta_1)/\theta_0]} \tag{13}$$

式中：θ_i 为隐含层节点的阈值。

隐含层第 i 个节点的输出 r_i 为：

$$r_i = f(\sum_{j=1}^{n}\omega_{ij}p_j+\theta_i) \tag{14}$$

式中：ω_{ij} 为节点 i 与节点 j 两者间的权值。

输出层第 k 个神经元的输出为：

$$\alpha_k = f(\sum_{i=1}^{s_i}\omega_{kj}r_i+b_k) \tag{15}$$

式中：b_k 为输出层节点 k 的阈值。

一般情况下，其输出值都是与实际期望有差异的，因此需要对其网络进行校正。采用的误差函数为：

$$E(\omega,b)=\frac{1}{2}\sum_{k=1}^{s_2}(t_k-b_k)^2 \tag{16}$$

输出层中从第 i 个输入值第 k 个的权值为：

$$\Delta\omega_{ki} = -\eta \frac{\partial E}{\partial \omega_{ki}} \tag{17}$$

式中：η 为学习速率。

隐含层中从第 j 个输入值第 i 个的权值为：

$$\Delta\omega_{ki} = -\eta \frac{\partial E}{\partial \omega_{ki}} \tag{18}$$

该网络是将输出的误差以相反的方向从输出层开始、经过隐含层、最后到达输入层，每一层都会做一个校正。为提高网络准确度，需要将 BP 网络中输入的所有样本做循环记忆训练，使得网络记住此样本模式。

我们首先采用单隐含层网络映射所有函数，如果一个隐含层的网络结构预测精度不够时，我们再增加隐含层数量。

采用问题一处理好的数据，将原料辛烷值(RON)等主要变量作为输入数据，产品辛烷值(RON)作为输出数据。从 321 个样本中随机选取 300 组样本数据作为网络训练数据，21 个样本数据作为网络测试数据，对训练数据进行归一化处理。其处理结果见表 4.16。

表 4.16 数据归一化处理结果

编码 / 样本	2	363	105	271	214	133	……	205
1	0.2500	-0.7987	0.4866	0.2653	-0.3399	-0.1352	……	-0.1555
2	0.4167	-0.1884	0.8312	-0.5998	0.5447	-0.7017	……	-0.4450
3	0.2083	-0.9337	0.3966	0.3688	-0.2211	-0.2279	……	0.0151
4	0.2500	0.0309	0.9615	-0.5505	0.5066	-0.1315	……	-0.0800
5	0.5000	0.0349	-0.3871	-0.0138	-0.1828	-0.1823	……	0.2515
6	-0.7500	0.0795	0.9907	-0.4063	0.3696	0.5934	……	-0.6743
7	0.0417	-0.4449	-0.6265	-0.3546	0.2148	0.1070	……	-0.2399
……	……	……	……	……	……	……	……	……
300	0.0417	-0.0515	0.9083	-0.7625	0.6610	-0.4860	……	0.0136

对于隐含层节点数设计，如果节点数过少，则学习的容量有限，不足以存储训练样本中蕴含的所有规律；而如果节点过多则会导致增加网络训练时间。因此采用凑试法，初步确定单隐含层的节点数为 9。

根据上述我们建立 BP 神经网络，其大致如下：

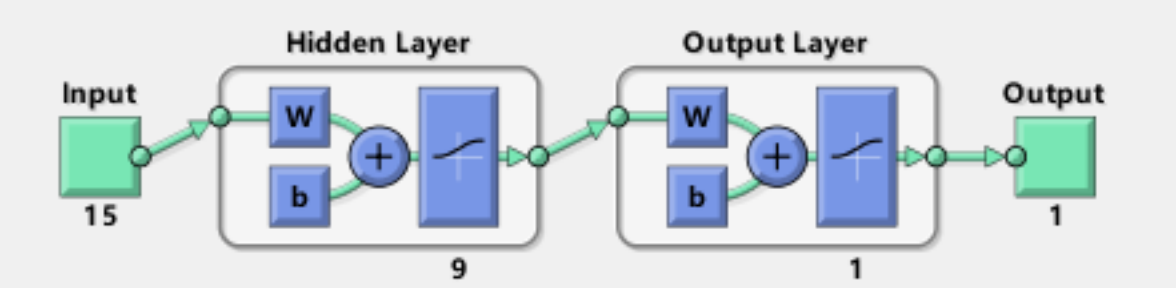

图 4.9 单隐含层 BP 神经网络结构示意图

预测结果和相关误差为：

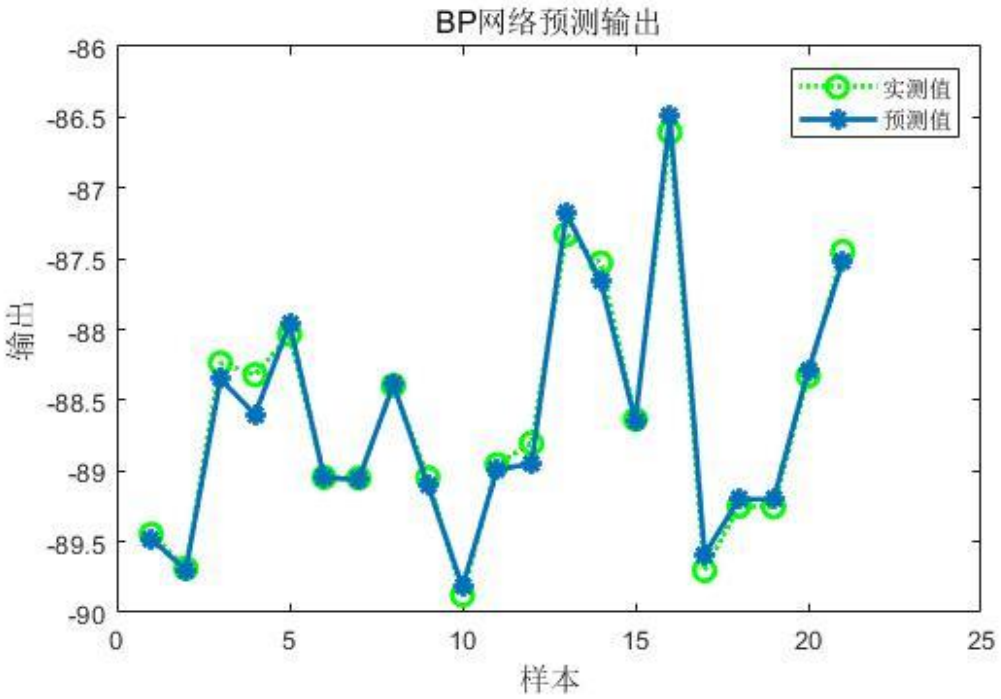

图 4.10 辛烷值实测值与单隐含层 BP 神经网络预测值对比

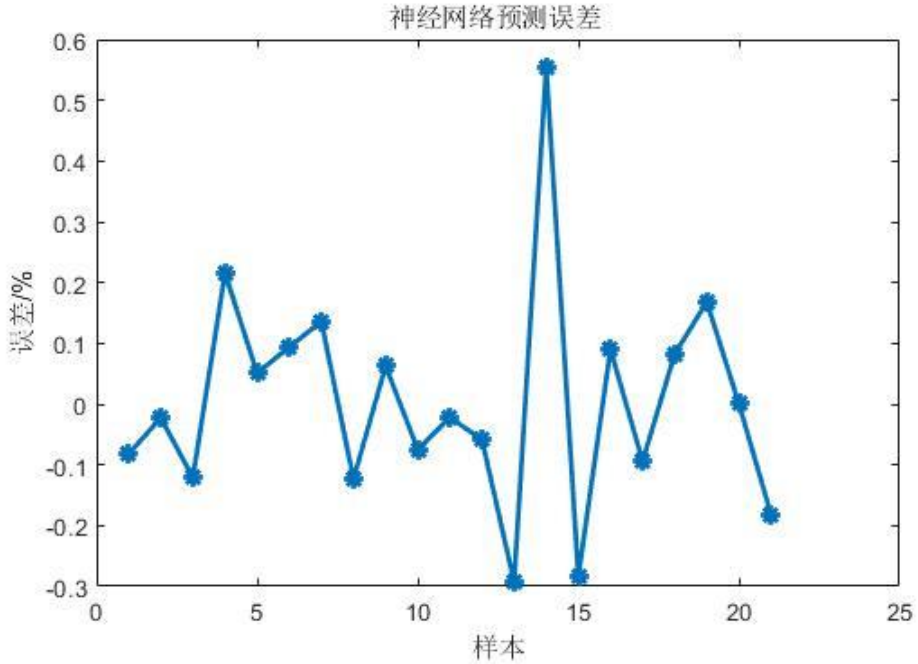

图 4.11 单隐含层 BP 神经网络辛烷值预测误差(%)

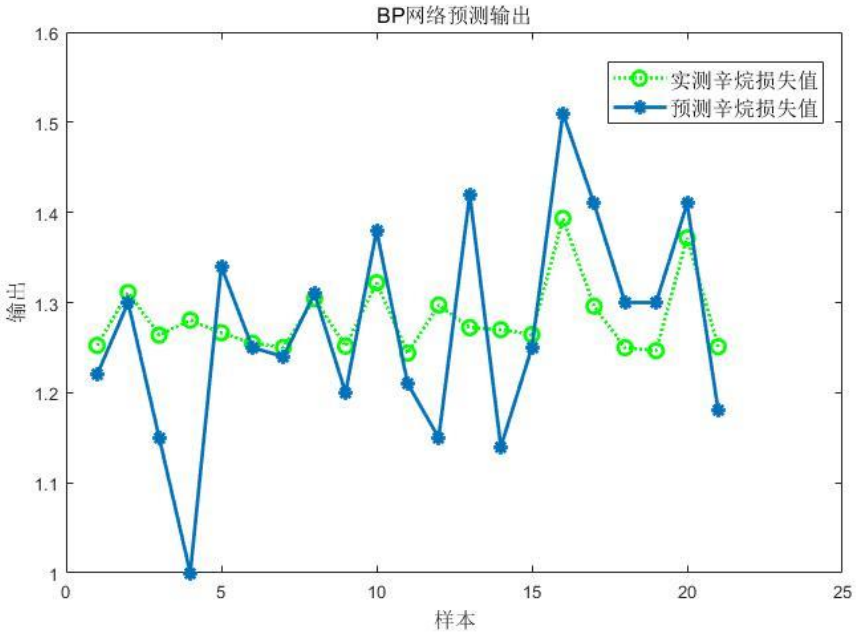

图 4.12 单隐含层 BP 神经网络辛烷值(RON)损失预测值与实测值误差

我们对于产品辛烷值的预测值与实测值的误差对比如图 4.9 与图 4.10 所示，其精度较高。但是对于辛烷值(RON)损失的预测仍存在约为 10~20%的误差，因此模型仍需要改进，提高其精度，采用双隐含层的神经网络，其网络结构如图 4.13 所示。

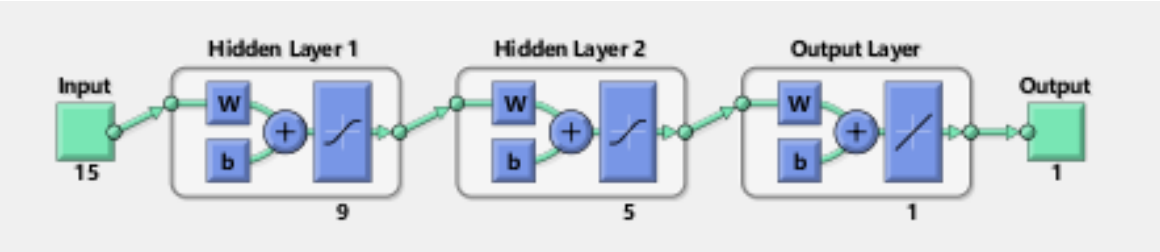

图 4.13 双隐含层 BP 神经网络结构示意图

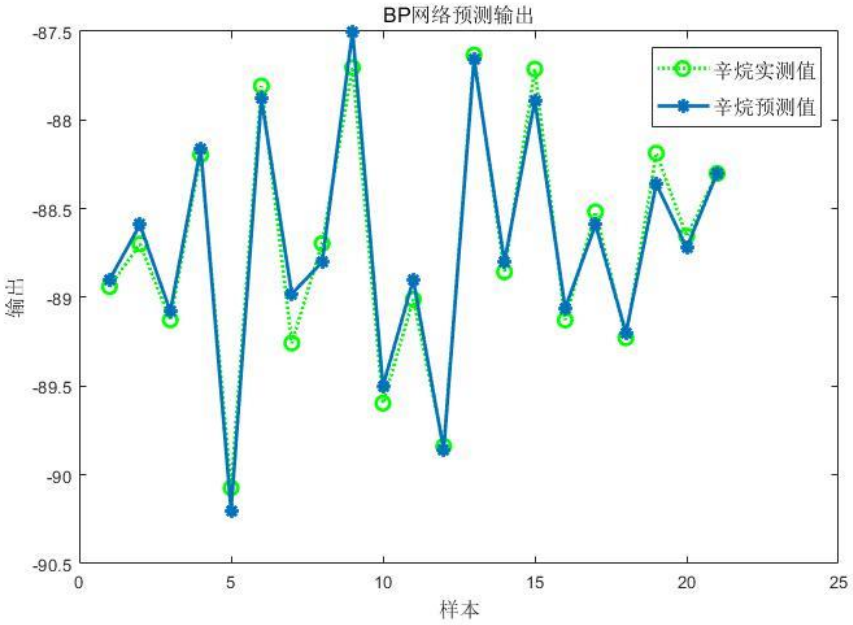

图 4.14 辛烷值实测值与双隐含层 BP 神经网络预测值对比

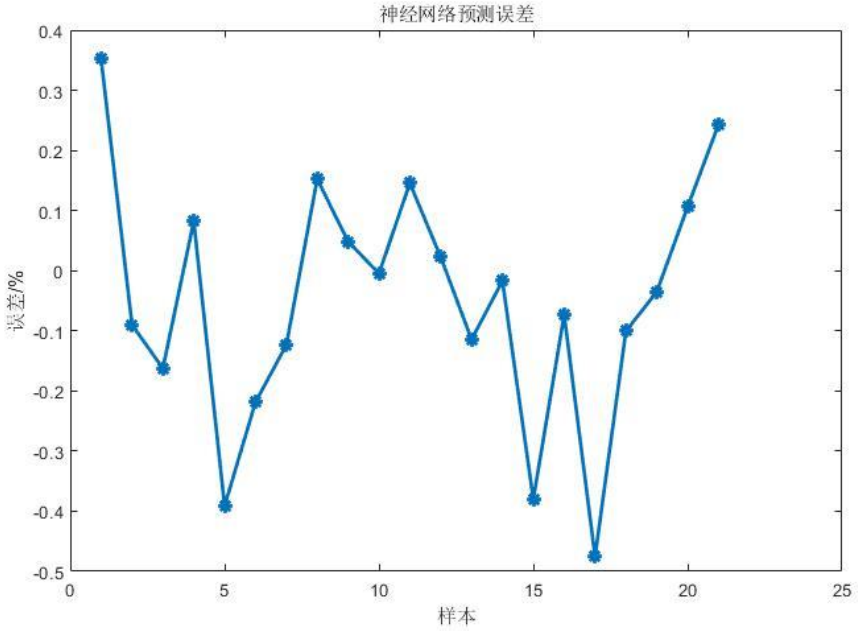

图 4.15 双隐含层 BP 神经网络辛烷值预测误差(%)

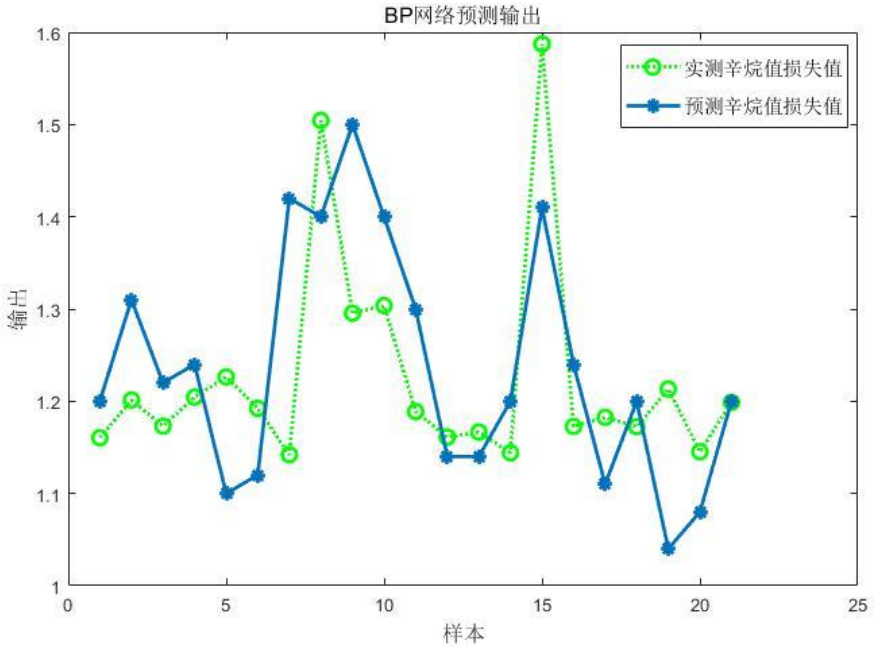

图 4.16 双隐含层 BP 神经网络辛烷值(RON)损失预测值与实测值误差

如图 4.14-4.16 所示，辛烷值预测值和实测值之间差距较小，采用双层神经网络预测的误差更加稳定，误差可以稳定在 0.5%以内，对于辛烷值(RON)损失的预测依然存在误差，但是实测值与预测值的整体变化趋势是相同的。为了进一步提高模型预测精度，我们对模型进行进一步的优化操作。

4.3.2 小波神经网络预测[4]

神经网络在处理非线性预测时表现出较好的应用性，为了对上述建立的模型进行优化，我们采用小波神经网络模型，将小波分析与神经网络相结合，在保持神经网络本身自学习、自适应等特点的同时增加多尺度观测的优势。

由于模型三的传递函数是 Sigmoid 函数，它在一定程度上存在重叠性，即它的基函数是有冗余的，因此其逼近函数的表达式并不唯一。因此我们采用嵌入式结合，用小波基函

数来替代神经网络 Sigmoid 函数，以解决神经网络陷入局部最优的不足。

设小波函数$\psi(t)\in L^2(R)$，其平均值为 0，若其傅里叶变换$\psi(w)$满足：

$$C_\psi=\int_{-\infty}^{+\infty}\frac{|\psi(w)|^2}{|w|}dw<+\infty \tag{19}$$

则ψ为基小波。我们采用的是 Morlet 小波，它是高斯包络下的单频率副正弦函数：

$$\psi(t)=\mathrm{C}\mathrm{e}^{\frac{t^2}{2}}\cos(5t) \tag{20}$$

式中：C 为重构时的归一化常数。通过将$\psi(t)$母函数经过伸缩和平移可以得到基小波函数族，即：

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{21}$$

式中：a、b分别为伸缩因子和平移因子。通过选择恰当的伸缩因子和平移因子对应输入层到隐含层的权值及阈值，使之具有更为灵活的函数逼近情况和模式容错能力。因此，在该网络中，隐含层输出计算公式为：

$$f(t)=\sum_{j\in Z}\omega_j\frac{1}{a_j}\psi(\frac{t-b_j}{a_j}) \tag{22}$$

我们采用梯度修正法修正网络的权值和小波基函数参数，以减小预测值与期望值的误差，其修正过程主要为：

（1）计算神经网络预测误差：

$$\Delta e=\sum_{k=1}^{m}yn(k)-y(k) \tag{23}$$

式中：$yn(k)$为期望输出；$y(k)$为小波神经网络预测输出。

（2）根据预测误差Δe的值，修正小波神经网络权值和小波基函数的系数：

$$w_{n,k}^{i+1}=w_{n,k}^{i}+\Delta w_{n,k}^{i+1} \tag{24}$$

$$a_{n,k}^{i+1}=a_{n,k}^{i}+\Delta a_{n,k}^{i+1} \tag{25}$$

$$b_{n,k}^{i+1}=b_{n,k}^{i}+\Delta b_{n,k}^{i+1} \tag{26}$$

式中：$\Delta W_{n,k}^{i+1}$、$\Delta a_{n,k}^{i+1}$、$\Delta b_{n,k}^{i+1}$是根据网络预测误差得到，即：

$$\Delta w_{n,k}^{i+1}=-\eta\frac{\partial\Delta e}{\partial w_{n,k}^{i}},\Delta a_{n,k}^{i+1}=-\eta\frac{\partial\Delta e}{\partial a_{n,k}^{i}},\Delta b_{n,k}^{i+1}=-\eta\frac{\partial\Delta e}{\partial b_{n,k}^{i}} \tag{27}$$

因此，我们可以建立小波神经网络的模型，主要步骤为：

（1）网络初始化。随机初始化伸缩因子a_k、平移因子b_k、节点间的连接权值ω_{ij}，设置学习速率η。

（2）样本分类。将筛选后的样本分为 300 个训练样本和 21 个测试样本，其中训练样本用于训练网络，测试样本用于测试网络预测精度。

（3）输出预测。将训练样本输入至网络中，计算网络的输出值，并将其与期望输出值对比计算其误差Δe。

（4）系数修正。根据误差修正权值和函数参数，从而使得预测值逼近期望值。

（5）判断算法终止条件，如果没有终止则返回步骤 3 再次计算。

采用小波预测结果如图 4.17、4.18 所示：

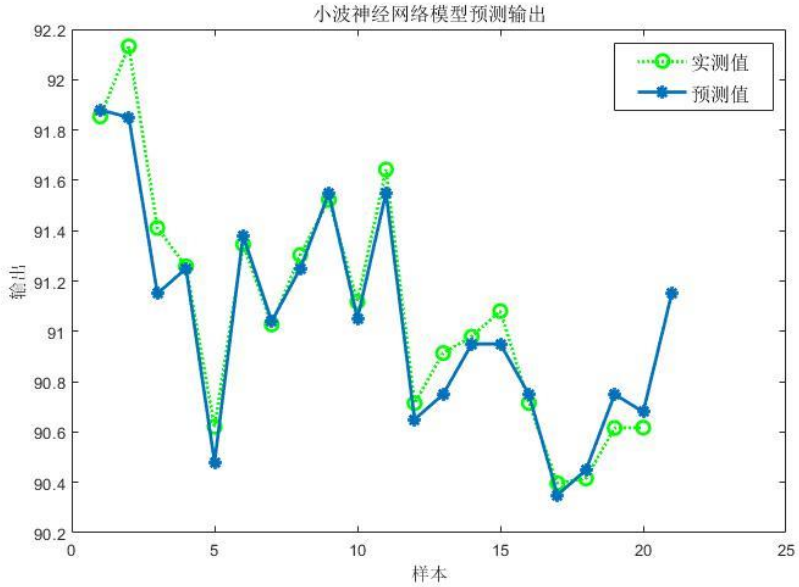

图 4.17 辛烷值实测值与小波神经网络预测值对比

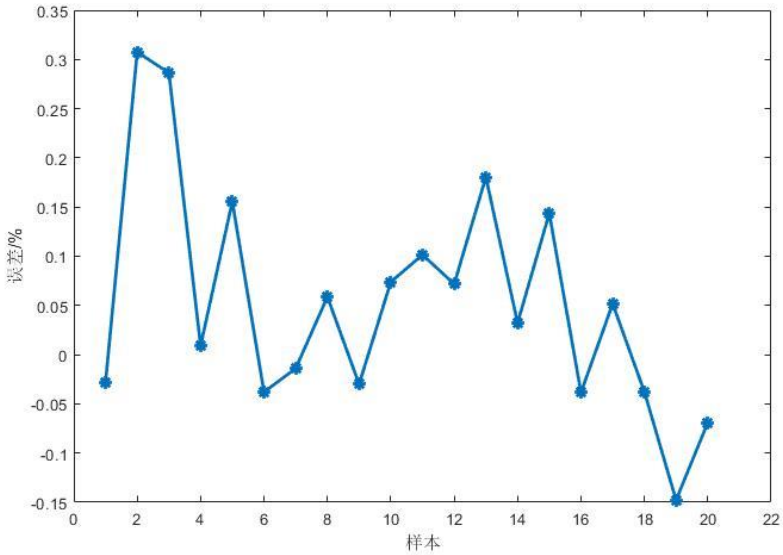

图 4.18 小波神经网络辛烷值预测误差(%)

由图 4.18 可知，采用小波神经网络优化之后，其预测精度误差在-0.15%~0.3%之间，对比单隐含层 BP 模型的预测误差在-0.36%~0.55%，双隐含层 BP 模型的预测误差在-0.4%~0.35%之间，可以看出小波神经优化后，其预测误差是有一定程度提高，但预测的精度整体上较 BP 神经网络有些许的提高，但是提高的幅度不大。

采用 BP 神经网络进行产品辛烷值的预测是可靠的，预测的精度较高，但是对于辛烷值(RON)损失的预测仍存在偏差，这主要是两方面的原因：(1)操作变量之间具有高度的非线性和相互强耦联，因此操作变量与产品辛烷值(RON)、辛烷值(RON)损失存在强烈的非线性关系，因此预测的精度有所损失；(2)所给的样本数据依然较少，因此要使得训练得到的神经网络预测精度进一步提高，必然需要更多的样本数据。

4.4 问题四：主要变量操作方案的优化

问题四要求在保证产品中硫含量不大于$5\mu g/g$的前提下，对降维后的主要操作变量进行优化，使得变量调整后的产品辛烷值(RON)损失比调整前的产品辛烷值(RON)损失降低超过 30%。题目中某石化企业的催化裂化汽油精制脱硫装置运行 4 年，积累了大量历史数

据，其汽油产品辛烷值(RON)损失平均为 1.37 个单位，而同类装置的最小损失值只有 0.6 个单位。因此问题四中，将辛烷值(RON)损失优化的合理目标区间定为（0.5,0.7）。在 15 个降维后的主要变量中，原料辛烷值(RON)属于原料的自身属性，优化过程不改变这一变量的取值。

4.4.1 模型五：基于 BP 神经网络的粒子群寻优模型[5]

根据问题四的相关分析，可以给出本问的目标函数 F ：

$$F=RON\text{损失}_{\text{调整值}}(\vec{x})-RON\text{损失}_{\text{目标值}}$$

其中，RON 损失调整函数取问题三中的 BP 神经网络模型，即 $RON\text{损失}_{\text{调整值}}(\vec{x})$，$\vec{x}$ 为降维后的主要变量的向量组合；$RON\text{损失}_{\text{目标值}}$ 为（0.5,0.7）。

附件四给出了各个操作变量的可调整范围，同时，汽油在精制过程中脱硫就必然导致产品中辛烷值小于原料中的辛烷值，故求解目标函数 F 最小值的约束条件为：

$$\begin{cases}\min_{x_i} < x_i < \max_{x_i}, i=1,2,3,\cdots,15\\ S \le 5\mu g/g\\ RON_{\text{调整值}} < RON_{\text{原料值}}\end{cases}$$

附件四还给出了各个操作变量单步调整的最大幅度Δ值，考虑到实际生产中操作变量调整的可执行性，依据假设 3、5，设定优化过程中的调整幅度等于单步调整的最大幅度Δ值；同时鉴于操作变量的采集频次为 1 次/3 分钟，且只有当前时刻的前两个小时的操作变量调整才会对产品辛烷值造成影响，因此，本问假定至多可进行 40 次调整。

另外，321 个样本中产品硫含量出现较多的异常值，例如 $3.2\mu g/g$ 出现多达 190 次，占比达到 59.2%，属于不正常现象，猜测是采集仪器的量程有限，最小量程数值为 $3.2\mu g/g$，同时考虑到题目要求只需要控制硫含量不超过 $5\mu g/g$，因此可以剔除这 190 个样本，再对硫含量进行预测，预测模型与问题三的模型三一致。这里仅仅展示预测的结果。观察预测结果可知，由于样本数量较少等客观因素，预测效果一般，所以在后续的模型中，对产品中硫含量这一约束予以加强，使得其尽量不超过 $4\mu g/g$，来弥补预测精度的不足。

通过该模型对硫含量进行预测，其硫含量的预测结果见图 4.19-4.20。

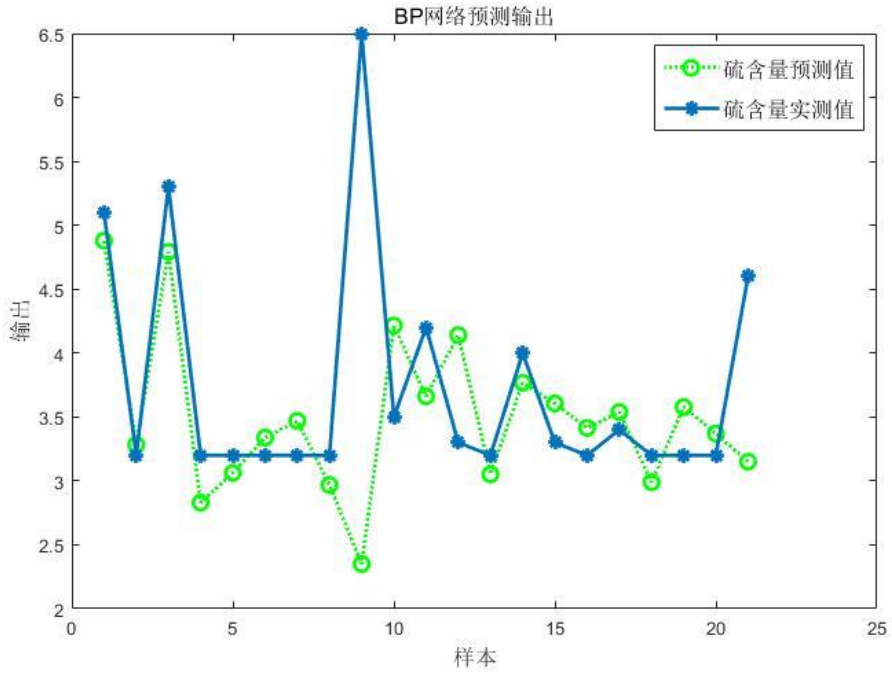

图 4.19 硫含量实测值与 BP 神经网络预测值对比

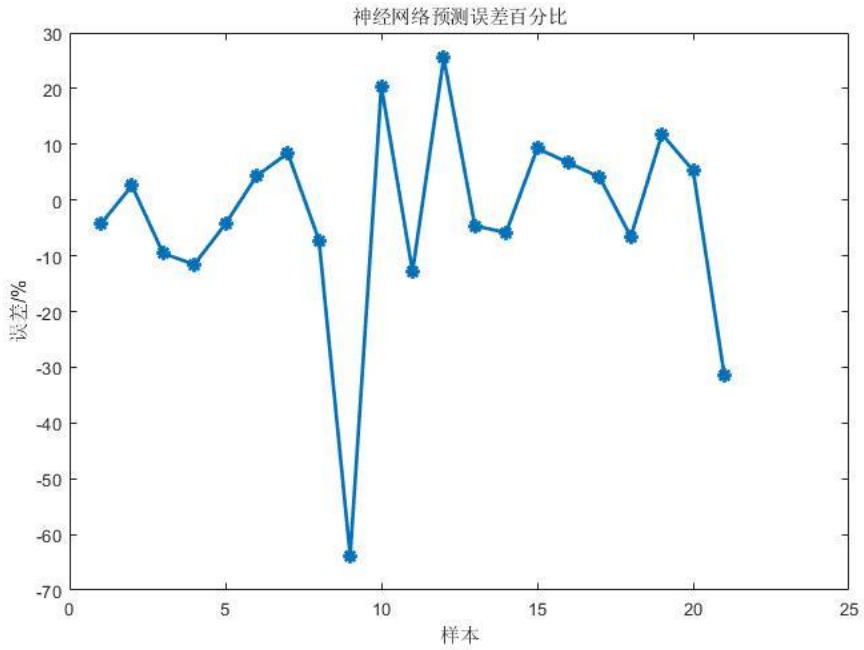

图 4.20 BP 神经网络硫含量预测误差(%)

本模型利用粒子群优化算法进行求解，粒子群算法（Particle Swarm Optimization，PSO）是一种基于群体的随机优化技术，它将每个可能产生的解表述为群中的每一个微粒，每个微粒都具有自己的位置向量和速度向量，以及一个由目标函数决定的适应度，所有微粒在搜索空间中以一定速度飞行，在初始化一组随机解后通过迭代搜寻最优解。

但是由于本问的特殊性，将粒子群算法作细微调整来解决该问题，现将求解过程给出：

（1）初始化一个群体规模 m=40 的粒子群，指定所有粒子的位置 p_{x_i} 为当前待调整样本中的变量取值，指定原料辛烷值(RON)这一变量所对应的粒子速度 v_{x_1} 为 0，以 50%的相对概

率随机初始化其他主要变量的速度 v_{x_i} 为 $\pm\Delta_{x_i}, i=2,3,\cdots,15$；

（2）计算每个粒子的适应度，若粒子对应的 *RON* 损失$_{调整值}(\vec{x})$ 超出目标范围或者产品中硫含量超过 $4\mu g/g$，该粒子适应度指定为较大值，其余情况按照目标函数 F 计算适应度；

（3）对每个粒子将其适应度和其经历过的最好位置 $p_{x_i}^{currentbest}$ 的适应度进行比较，若较好，则将其作为当前的最好位置；

（4）对每个粒子将其适应度和全局经历过的最好位置 $p_{x_i}^{globalbest}$ 的适应度进行比较，若较好，则将其作为当前的全局最好位置；

（5）对粒子的速度进行更新，若 $rand(1,1)>0.5$，则 $v_{x_i}=\Delta_{x_i}$，若 $rand(1,1)<0.5$，则 $v_{x_i}=-\Delta_{x_i}$；

（6）对粒子的位置进行更新，$p_{x_i}=p_{x_i}+v_{x_i}$；

（7）如果达到设定的最大迭代次数 40，则输出解，否则返回步骤（2）。

算法流程如图 4.21 所示。

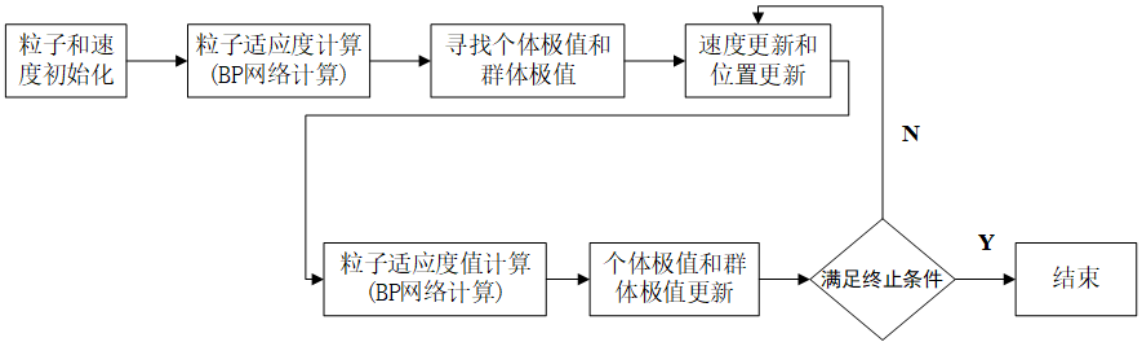

图 4.21 粒子群算法流程图

4.4.2 模型结果

采用模型求解得到主要变量的最优操作条件，见表 4.17。

表 4.17 主要变量的最优操作条件

编码 / 样本	2	363	105	271	214	133	……	205
1	90.600	27.821	871.817	0.416	448.457	27.713	……	39568757.000
2	90.500	37.005	678.941	26.104	439.138	32.454	……	39409299.000
3	90.700	34.811	663.586	21.125	434.632	31.544	……	39292616.500
4	90.400	40.002	931.971	43.666	429.963	24.271	……	39170204.500
5	89.600	27.444	564.347	20.929	428.323	29.114	……	39015953.500
6	91.000	40.790	531.153	14.720	432.741	18.729	……	38850284.500
7	90.400	30.328	174.437	2.916	428.501	23.889	……	38787013.000
……	……	……	……	……	……	……	……	……
321	89.900	38.795	1644.517	12.804	442.047	40.934	……	515996.800

4.5 问题五：模型的可视化展示

工业装置为了平稳生产，优化后的主要操作变量往往只能逐步调整到位。问题四已经做出假设，操作变量单步调整的幅度取对应的 Δ_{x_i} 值，且每个操作变量最多调整 40 次。利用问题四模型对 133 号样本进行操作变量优化。

给出优化过程产品中汽油辛烷值 RON 和硫含量的变化轨迹以及对应的操作变量的调整过程。133 号样本初始的原料辛烷值 RON 为 89.40 个单位，产品汽油辛烷值 RON 为 88.09 个单位，初始产品硫含量为 $3.20\mu g/g$，优化调整后的汽油辛烷值 RON 为 88.71 个单位，

辛烷值(RON)损失降幅达到 52.67%，辛烷值 RON 损失值为 0.69 个单位，达到同类装置的最优区间，优化调整后的产品硫含量为 $2.54\mu g/g$，且中间优化过程硫含量并未超出 $5.00\mu g/g$，满足控制要求。图 4.23 清晰地给出了操作变量的调整过程，红色方块符号代表在对应调整次数给原有操作变量取值加一个单位的 Δ_{x_i}，绿色方块符号代表在对应调整次数给原有操作变量取值减一个单位的 Δ_{x_i}，黑色方块符号代表不对操作变量作任何处理。

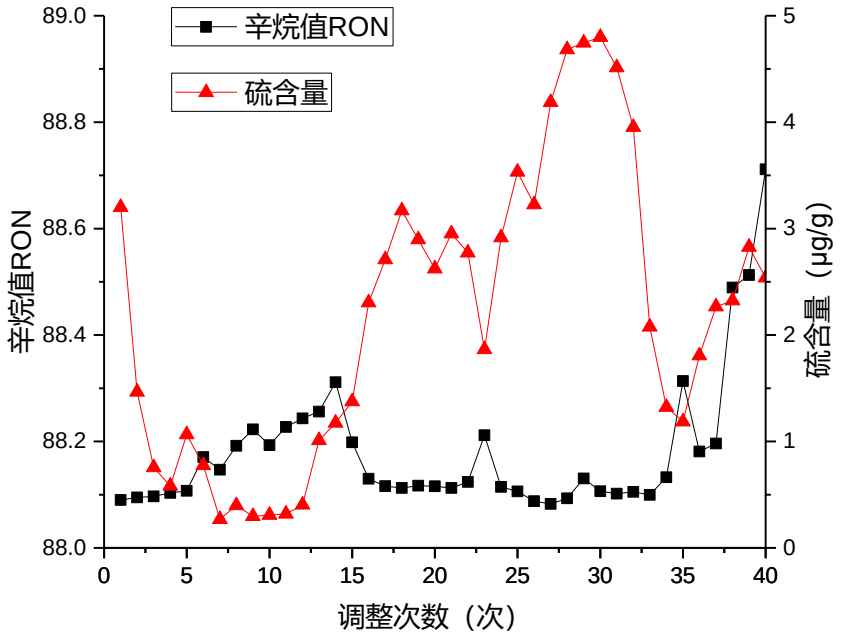

图 4.22 优化过程中汽油辛烷值和硫含量的变化轨迹

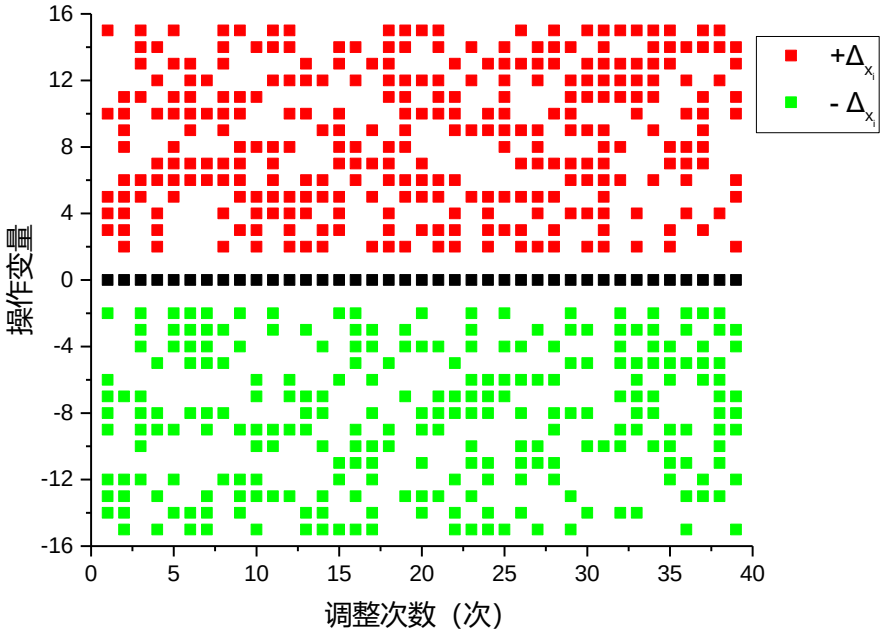

图 4.23 优化过程中操作变量的调整过程

五、 模型检验与评价

5.1 模型的检验

对于问题三和四中的模型，分别验证产品辛烷值(RON)和辛烷值(RON)损失，在题设区间里，验证了模型结果的合理性。

5.2 模型的优点

对于问题二，我们设计了基于互信息的非线性相关分析模型和基于遗传算法-BP 神经网络的降维模型，两个模型筛选出来的变量存在一定程度上的重合，并且选出来的主要变量具有代表性和独立性。基于遗传算法-BP 神经网络的降维模型选择的变量，其与因变量的相关度相对于前者选择的变量较低一些。因此选择基于互信息的非线性相关分析模型所挑出来的 15 个主要变量，通过这两个模型的对比，确保了我们所选择的主要变量符合题目所提出的代表性、独立性以及合理性的要求。

对于问题三，我们采用了 BP 神经网络预测模型对产品辛烷值(RON)进行预测，通过增加隐含层数量和小波神经网络函数优化预测模型，三种优化模型的计算结果显示双隐含层 BP 神经网络预测模型预测结果更好，其误差精度绝对值可以达到 0.48%以内。因此本文采用双隐含层 BP 神经网络预测模型作为产品辛烷值(RON)预测的模型，将预测的产品辛烷值(RON)和原料辛烷值(RON)计算就可以得到辛烷值损失预测值。

对于问题四，通过粒子群寻优算法对问题三建立的双隐含层 BP 神经网络预测模型进行优化，建立了基于双隐含层 BP 神经网络的粒子群寻优模型。该模型算法复杂度较高，对于模型的优化程度较高，提高了遍寻主要变量最优操作条件的准确性及合理性。

对于问题五，在问题四建立的基础上，通过逐步调整主要变量的取值，在限制硫含量的前提下，可视化产品辛烷值(RON)的变化轨迹，体现了模型对实际工艺的吻合度。

5.3 模型的缺点

对于问题二，模型暂未仔细考虑主要变量的个数对产品辛烷值(RON)预测精度的影响，以及基于遗传算法—BP 神经网络的降维模型由于计算量较大的原因没有将所有变量放入计算，对于变量的选择可能不是最优的。

对于问题三，对于预测模型的优化没有达到最理想的情况且受上一个问题变量选择的准确性，所以可能存在误差。

对于问题四，通过主要变量的逐步调整预测产品辛烷值(RON)损失，没有建立具象的拟合关系。

六、 模型推广

问题二的模型严格满足题目条件，可以适用于高维样本数据的变量筛选情况，对于化工工艺建模有较高参考性。

问题三和四的模型复杂度较高，对于变量较多、无法直接给出明确函数关系的问题，可以进行深入研究和应用。

七、 结论

本文充分利用线性、非线性相关分析、互信息理论、神经网络、智能寻优算法等数据挖掘技术研究了如何降低汽油精制过程中辛烷值损失的问题。

针对问题一，依据“样本确定方法”完成数据整定，得到 321 个样本、364 个变量的数据文件。

针对问题二，利用简单线性相关分析和偏相关分析，得到原料辛烷值(RON)在内的 15 个主要变量；基于遗传算法—BP 神经网络的降维模型得到最优的 14 个主要变量的组合。对比发现模型一更符合主要变量的独立性标准。

针对问题三，基于 15 个主要变量，建立 BP 神经网络模型预测产品辛烷值(RON)。选定双隐含层 BP 神经网络作为最终的预测模型。

针对问题四，建立基于 BP 神经网络预测模型的粒子群寻优算法，假定合理的产品辛烷值目标区间为（0.5,0.7），最终优化得到全部 325 个样本的最优操作条件。

针对问题五，可视化产品辛烷值(RON)和硫含量的变化轨迹，并给出具体的操作变量轨迹。结果表明：133 号样本的产品辛烷值(RON)由初始的 88.09 个单位提高到 88.71 个单位，辛烷值损失降幅达到 52.67%，同时，调整过程中硫含量始终不超过。

八、 参考文献

[1] 张凯. 高维小样本数据的互信息特征选择方法研究 [D]; 山西大学, 2017.

[2] 俞颖, 黄风华, 刘永芬. 基于特征降维及参数优化的语音情感识别 延边大学学报(自然科学版) [J]. 2020, 46(01): 49-54.

[3] 王增平, 杨国生, 汤涌, et al. 基于特征影响因子和改进 BP 算法的直驱风机风电场建模方法 中国电机工程学报 [J]. 2019, 39(09): 2604-15.

[4] 王旭. 改进型小波神经网络预报卫星钟差;第十七届沈阳科学学术年会, 中国辽宁沈阳, F, 2020 [C].

[5] 秦琪怡, 郭承湘, 吴帅, et al. 基于粒子群和布谷鸟搜索的 BP 神经网络优化方法研究 广西大学学报(自然科学版) [J]. 2020, 45(04): 898-905.

附录

问题一：数据处理主要程序

```
% 本程序主要用于问题一的数据处理
clc,clear;
data1=textread('11.txt');
d31=xlsread('3-1.xlsx');
d32=xlsread('3-2.xlsx');
d11=xlsread('1-1.xlsx');
dz=xlsread('zdata.xlsx');
d3leng=size(d31,2);
d1leng=size(d11,1);
d3leng2=size(d31,1);
dele=[];
……
% 将 285 数据以及 313 数据存储至附件 1 数据中
wu=zeros(2,d3leng);
for i=1:d3leng
    wu(1,i)=d11(285,i)-chuli(i);
    wu(2,i)=d11(313,i)-chuli2(i);
end
for i=1:d3leng
    if ~isnan(chuli(i))
        d11(285,i)=chuli(i);
    end
    if ~isnan(chuli2(i))
        d11(313,i)=chuli2(i);
    end
end

[m1 n1]=size(dz);
% 将非操作变量进行 3sigema 处理
cun1=[];
sig=[];
xba1=[];
des=zeros(m1,14);
for i=1:14
    sum=0;
    num=0;
    cun=[];
    sum2=0;
    znum=0;
    zba=0;
    sum3=0;
    for j=1:m1
        sum=sum+dz(j,i);
        num=num+1;
        cun=[cun dz(j,i)];
```

```
    end
    xba=sum/num;
    xba1=[xba1 xba];
    cun=cun-xba;
    for k=1:num
        sum2=sum2+cun(k)^2;
    end
    sigema=sqrt(sum2/(num-1));
    sig=[sig sigema];
    for j=1:m1
        if abs(cun(j))<=3*sigema
            des(j,i)=1;
        end
    end
end


% 将关于原料辛烷值、产品辛烷值、和辛烷损失值超过 3sigema 的样本删除
del1(:,1)=des(:,2);
del1(:,2)=des(:,9);
del1(:,3)=des(:,10);
del2(:)=del1(:,1)+del1(:,2)+del1(:,3);
j=0;
cc1=[];
for i=1:d1leng
    if del2(i)==3
        j=j+1;
        aa(j,:)=dz(i,:);
    else
        cc1=[cc1 i];
    end
end

% 将数据异常的 22、49、84、111 列删除
bb=aa;
bb(:,[36 63 98 125])=[];
% 输出整定后的数据
xlswrite('321.xlsx',bb);
```

问题二：模型一主要程序

```
% 本程序用于挑选计算所用的变量
clc,clear
ppz=xlsread('321.xlsx');
pp2=xlsread('bzh.xlsx');
% 将数据进行从新处理，将产品辛烷值、硫、损伤值作为 1,2,3,存储于 pp1 中
m1=size(ppz,1);
```

```
m2=size(ppz,2);
pp1(:,1)=ppz(:,9);
pp1(:,2)=ppz(:,8);
pp1(:,3)=ppz(:,10);
pp1(:,4:10)=ppz(:,1:7);
pp1(:,11:m2)=ppz(:,11:m2);

% 选择三个线性先关系数强的变量
cc=corrcoef(pp1);
dd=abs(cc(1,:));
nn1=100;
maxx=[];
for i=1:nn1
    [m,p1]=max(dd);
    maxx=[maxx; m p1];
    dd(p1)=0;
end
% 保证选择的变量之间非强相关

% 数据标准化处理
pp=zeros(m1,m2-3);
 for i=1:m2
    pp(:,i)=zscore(pp1(:,i));
 end

p=pp(:,4:m2);
t1=pp1(:,1);
t2=pp1(:,2);
t3=pp1(:,3);
t1=(t1-min(t1))/(max(t1)-min(t1));%zscore(t1);
t2=zscore(t2);
t3=zscore(t3);
u1 = t1;
for k=4:m2
    %p(:,k)=zscore(p(:,k));
    u2 = pp1(:,k);
    wind_size = size(u1,1);
    mi = calmi(u1, u2, wind_size);
    MIC(k)=mi;
end

mic2=MIC;
% 寻找最大值
max1=[];
nu1=m2-3;
for i=1:nu1
    [m,p1]=max(mic2);
    max1=[max1; m p1];
    mic2(p1)=0;
```

```
end
% 按照互相关系数大小输出数据
xlswrite('max1.xlsx',max1);

% 互相关数据的关联性
for i=1:m2-3
    gl(:,i)=pp1(:,max1(i,2));
end
gl1=corrcoef(gl);
xlswrite('guanlian.xlsx',gl1);

nu=90; %数据个数
a1=zeros(m1,nu);
q1=1;
q2=2;
max2=max1(1,:);
aa=pp1(:,max2(1,2));
a1=[];
for i=1:nu
    j=0;
    pan1=1;
    while pan1>=0.8
        j=j+1;
        q2=q1+j;
        s1=pp(:,max1(q1,2));
        bb=aa;
        s2=pp(:,max1(q2,2));
        bb=[bb s2];
        length=size(bb,2);
        des=corrcoef(bb);
        % 使得变量之间的相关系数尽可能大
        pan = corrcoef(s1,s2);
        % pan1=abs(pan(1,2));
        pan1=0;
        for k=1:length-1
            if des(length,k)>0.5
                pan1=1+pan1;
            end
        end
    end
    q1=q2;
    max2=[max2;max1(q2,1) max1(q2,2)];
    aa=[aa pp1(:,max1(q2,2))];
end
xlswrite('max22.xlsx',max2);

% 将样本数据输出
for i=1:nu
    a1(:,i)=pp1(:,max2(i,2));
```

```
end

a2=zeros(m1,nu+3);
a2(:,1:3)=pp1(:,1:3);
a2(:,4)=pp1(:,maxx(2,2));
a2(:,5)=pp1(:,maxx(3,2));
a2(:,6)=pp1(:,maxx(4,2));

a2(:,7:nu+6)=a1(:,:);
xlswrite('data2.xlsx',a2);

pn1=15;
% 输出第四问所需数据
b1(:,1)=pp1(:,3);
b1(:,2)=pp1(:,2);
b1(:,3:pn1+2)=a1(:,1:pn1);
xlswrite('question4.xlsx',b1);

% 输出遗传算法所需的数据
a3(:,1:3)=pp1(:,1:3);
for i=1:3
    a3(:,i+3)=pp1(:,maxx(i+1,2));
end
a3(:,7:nu+6)=a1(:,:);
xlswrite('data3.xlsx',a3);
```

问题二：模型二程序

```
% 本程序用于 BP 模型的数据读取
clc
clear
pp=xlsread('BP_data.xlsx');
input=pp(:,4);
input=[input pp(:,7:20)];
t1=pp(:,1);
t2=pp(:,2);
t3=pp(:,3);
%t1=(t1-min(t1))/(max(t1)-min(t1));%zscore(t1);
output = t3;

save data1 input output


% 基于遗传算法-BP 神经网络的降维模型
%% 遗传算法的优化计算——输入自变量降维

%% 清空环境变量
clear all
clc
```

```
warning off
%% 声明全局变量
global P_train T_train P_test T_test    mint maxt S s1
aa=28;
S = aa;
s1 = aa;
%% 导入数据
load data.mat
a = randperm(321);
Train = data(a(1:300),:);
Test = data(a(301:end),:);
% 训练数据
P_train = Train(:,4:aa+3)';
T_train = Train(:,1)';
% 测试数据
P_test = Test(:,4:aa+3)';
T_test = Test(:,1)';

%% 数据归一化
[P_train,minp,maxp,T_train,mint,maxt] = premnmx(P_train,T_train);
P_test = tramnmx(P_test,minp,maxp);
%% 创建单 BP 网络
t = cputime;
net_bp = newff(minmax(P_train),[s1,1],{'tansig','purelin'},'trainlm');
% 设置训练参数
net_bp.trainParam.epochs = 1000;
net_bp.trainParam.show = 10;
net_bp.trainParam.goal = 0.1;
net_bp.trainParam.lr = 0.1;
net_bp.trainParam.showwindow = 0;
%% 训练单 BP 网络
net_bp = train(net_bp,P_train,T_train);
%% 仿真测试单 BP 网络
tn_bp_sim = sim(net_bp,P_test);
% 反归一化
T_bp_sim = postmnmx(tn_bp_sim,mint,maxt);
e = cputime - t;
result_bp = [T_bp_sim' T_test'];
%% 结果显示（单 BP 网络）
disp(['建模时间为：' num2str(e) 's'] );
%% 遗传算法优化
popu = 20;
bounds = ones(S,1)*[0,1];
% 产生初始种群
initPop = randi([0 1],popu,S);
% 计算初始种群适应度
initFit = zeros(popu,1);
for i = 1:size(initPop,1)
```

```
    initFit(i) = de_code(initPop(i,:));
end
initPop = [initPop initFit];
gen = 100;
% 优化计算
[X,EndPop,BPop,Trace] = ga(bounds,'fitness',[],initPop,[1e-6 1 0],'maxGenTerm',...
    gen,'normGeomSelect',0.09,'simpleXover',2,'boundaryMutation',[2 gen 3]);
[m,n] = find(X == 1);
disp(['优化筛选后的输入自变量编号为:' num2str(n)]);
% 绘制适应度函数进化曲线
figure
plot(Trace(:,1),Trace(:,3),'r:')
hold on
plot(Trace(:,1),Trace(:,2),'b')
xlabel('进化代数')
ylabel('适应度函数')
title('适应度函数进化曲线')
legend('平均适应度函数','最佳适应度函数')
xlim([1 gen])
%% 新训练集/测试集数据提取
p_train = zeros(size(n,2),size(T_train,2));
p_test = zeros(size(n,2),size(T_test,2));
for i = 1:length(n)
    p_train(i,:) = P_train(n(i),:);
    p_test(i,:) = P_test(n(i),:);
end
t_train = T_train;
%% 创建优化 BP 网络
t = cputime;
net_ga = newff(minmax(p_train),[s1,1],{'tansig','purelin'},'trainlm');
% 训练参数设置
net_ga.trainParam.epochs = 1000;
net_ga.trainParam.show = 10;
net_ga.trainParam.goal = 0.1;
net_ga.trainParam.lr = 0.1;
net_ga.trainParam.showwindow = 0;
%% 训练优化 BP 网络
net_ga = train(net_ga,p_train,t_train);
%% 仿真测试优化 BP 网络
tn_ga_sim = sim(net_ga,p_test);
% 反归一化
T_ga_sim = postmnmx(tn_ga_sim,mint,maxt);
e = cputime - t;
result_ga = [T_ga_sim' T_test'];
%% 结果显示（优化 BP 网络）
disp(['建模时间为：' num2str(e) 's'] );
error1=(result_bp(:,1)-result_bp(:,2))./result_bp(:,2);
error2=(result_ga(:,1)-result_ga(:,2))./result_ga(:,2);
```

问题三：模型三程序

```
%  本程序用于 BP 模型的数据读取
clc
clear
pp=xlsread('BP_data.xlsx');
input=pp(:,4);
input=[input pp(:,7:20)];
t1=pp(:,1);
t2=pp(:,2);
t3=pp(:,3);
output = t3;

save data1 input output
%%  本程序主要用于 BP 神经网络
clc
clear

tic
%%  训练数据预测数据提取及归一化
%下载输入输出数据
load data1 input output

errorsum=10;
err1=10;
j=0;
ee1=[];
while errorsum>=1.5&&j<=100
%从 1 到 321 间随机排序
ss=321;
  k=rand(1,ss);
  [m,n]=sort(k);
%找出训练数据和预测数据
j=j+1;
t1=300;
input_train=input(n(1:t1),:)';
output_train=output(n(1:t1),:)';
input_test=input(n(t1+1:ss),:)';
output_test=output(n(t1+1:ss),:)';

%选连样本输入输出数据归一化
[inputn,inputps]=mapminmax(input_train);
[outputn,outputps]=mapminmax(output_train);

%% BP 网络训练
% %初始化网络结构
net=newff(inputn,outputn,[9 5]);
```

```
net.trainParam.epochs=100;
net.trainParam.lr=0.1;
net.trainParam.goal=0.0000004;

%网络训练
net=train(net,inputn,outputn);

%% BP 网络预测
%预测数据归一化
inputn_test=mapminmax('apply',input_test,inputps);

%网络预测输出
an=sim(net,inputn_test);

%网络输出反归一化
BPoutput=mapminmax('reverse',an,outputps);

%预测误差
error=BPoutput-output_test;
errorsum=sum(abs(error));
ee1=[ee1 errorsum];
if errorsum<err1
    err1=errorsum;
    BPoutput1=BPoutput;
    output_test1=output_test;
    input_test1=input_test;
    input_train1=input_train;
end
end

%% 结果分析
eva1=mean(ee1);
BPoutput=BPoutput1;
output_test=output_test1;
input_test=input_test1;
input_train=input_train1;
yuanliao=input_test(1,:);
k1=BPoutput-yuanliao;
k2=output_test-yuanliao;
figure(1)
plot(BPoutput,':og')
hold on
plot(output_test,'-*');
%legend('预测输出','期望输出','fontsize',12)
title('BP 网络预测输出','fontsize',12)
xlabel('样本','fontsize',12)
ylabel('输出','fontsize',12)

figure(2)
```

```
plot(error,'-*')
title('神经网络预测误差')

figure(3)
plot((BPoutput./output_test-1)*100,'-*');
title('神经网络预测误差百分比')

figure(4)
plot((k1./k2-1)*100,'-*');
title('神经网络预测误差百分比')

figure(5)
plot(k1,':og')
hold on
plot(k2,'-*');
%legend('预测输出','期望输出','fontsize',12)
title('BP 网络预测输出','fontsize',12)
xlabel('样本','fontsize',12)
ylabel('输出','fontsize',12)

toc

% 将数据集分开
xlswrite('input_test3.xlsx',input_test);
xlswrite('input_train3.xlsx',input_train);

save data net inputps outputps
```

问题三：模型四程序

```
% 本程序主要用于小波神经网络数据的读入
clc
clear
pp=xlsread('data321.xlsx');
input=pp(8:275,2:16);
output = pp(8:275,1);
input_test=pp(276:296,2:16);
output_test = pp(276:296,1);
save data input output input_test output_test

%% 该代码用于小波神经网络的预测
clc
clear

%% 网络参数配置
load data input output input_test output_test

M=size(input,2); %输入节点个数
```

```
N=size(output,2); %输出节点个数

n=6; %隐形节点个数
lr1=0.01; %学习概率
lr2=0.001; %学习概率
maxgen=1000; %迭代次数

%权值初始化
Wjk=randn(n,M);Wjk_1=Wjk;Wjk_2=Wjk_1;
Wij=randn(N,n);Wij_1=Wij;Wij_2=Wij_1;
a=randn(1,n);a_1=a;a_2=a_1;
b=randn(1,n);b_1=b;b_2=b_1;

%节点初始化
y=zeros(1,N);
net=zeros(1,n);
net_ab=zeros(1,n);

%权值学习增量初始化
d_Wjk=zeros(n,M);
d_Wij=zeros(N,n);
d_a=zeros(1,n);
d_b=zeros(1,n);

%% 输入输出数据归一化
[inputn,inputps]=mapminmax(input');
[outputn,outputps]=mapminmax(output');
inputn=inputn';
outputn=outputn';

error=zeros(1,maxgen);

%% 网络训练
for i=1:maxgen
    %误差累计
    error(i)=0;

    % 循环训练
    for kk=1:size(input,1)
        x=inputn(kk,:);
        yqw=outputn(kk,:);

        for j=1:n
            for k=1:M
                net(j)=net(j)+Wjk(j,k)*x(k);
                net_ab(j)=(net(j)-b(j))/a(j);
            end
            temp=mymorlet(net_ab(j));
```

```
    for k=1:N
        y=y+Wij(k,j)*temp;      %小波函数
    end
end

%计算误差和
error(i)=error(i)+sum(abs(yqw-y));

%权值调整
for j=1:n
    %计算 d_Wij
    temp=mymorlet(net_ab(j));
    for k=1:N
        d_Wij(k,j)=d_Wij(k,j)-(yqw(k)-y(k))*temp;
    end
    %计算 d_Wjk
    temp=d_mymorlet(net_ab(j));
    for k=1:M
        for l=1:N
            d_Wjk(j,k)=d_Wjk(j,k)+(yqw(l)-y(l))*Wij(l,j) ;
        end
        d_Wjk(j,k)=-d_Wjk(j,k)*temp*x(k)/a(j);
    end
    %计算 d_b
    for k=1:N
        d_b(j)=d_b(j)+(yqw(k)-y(k))*Wij(k,j);
    end
    d_b(j)=d_b(j)*temp/a(j);
    %计算 d_a
    for k=1:N
        d_a(j)=d_a(j)+(yqw(k)-y(k))*Wij(k,j);
    end
    d_a(j)=d_a(j)*temp*((net(j)-b(j))/b(j))/a(j);
end

%权值参数更新
Wij=Wij-lr1*d_Wij+k*(Wij_1-Wij_2);
Wjk=Wjk-lr1*d_Wjk+k*(Wjk_1-Wjk_2);
b=b-lr2*d_b+k*(b_1-b_2);
a=a-lr2*d_a+k*(a_1-a_2);

d_Wjk=zeros(n,M);
d_Wij=zeros(N,n);
d_a=zeros(1,n);
d_b=zeros(1,n);

y=zeros(1,N);
net=zeros(1,n);
net_ab=zeros(1,n);
```

```
            Wjk_1=Wjk;Wjk_2=Wjk_1;
            Wij_1=Wij;Wij_2=Wij_1;
            a_1=a;a_2=a_1;
            b_1=b;b_2=b_1;
        end
end

%% 网络预测
%预测输入归一化
x=mapminmax('apply',input_test',inputps);
x=x';
yuce=zeros(21,1);
%网络预测
for i=1:21
        x_test=x(i,:);

        for j=1:1:n
            for k=1:1:M
                net(j)=net(j)+Wjk(j,k)*x_test(k);
                net_ab(j)=(net(j)-b(j))/a(j);
            end
            temp=mymorlet(net_ab(j));
            for k=1:N
                y(k)=y(k)+Wij(k,j)*temp ;
            end
        end

        yuce(i)=y(k);
        y=zeros(1,N);
        net=zeros(1,n);
        net_ab=zeros(1,n);
end
%预测输出反归一化
ynn=mapminmax('reverse',yuce,outputps);
yun2=input_test(:,1);

%% 结果分析
figure(1)
plot(yun2+ynn,':og')
hold on
plot(yun2+output_test,'-*')
title('小波神经网络模型预测输出','fontsize',12)
%legend('预测交通流量','实际交通流量','fontsize',12)
xlabel('样本')
ylabel('输出')
error=((yun2+ynn)./(yun2+output_test)-1)*100;
figure(2)
plot(error,'-*')
```

```
figure(3)
plot(ynn,':og')
hold on
plot(output_test,'-*')
title('小波神经网络模型预测输出','fontsize',12)
%legend('预测交通流量','实际交通流量','fontsize',12)
xlabel('样本')
ylabel('输出')
```

问题四：模型五主要程序

```
%% 该代码为基于变异粒子群算法的函数极值寻优算法
global MMM
%% 参数初始化
%粒子群算法中的两个参数
%c1 = 1.19445;
%c2 = 1.19445;

maxgen=40;     % 进化次数
sizepop=70;     %种群规模

%Vmax=1;
%Vmin=-1;
load data input
load data1 net1 inputps outputps1
mm=xlsread('maxmin.xlsx');
popmax1=mm(2,:);
popmin1=mm(1,:);
delta=mm(3,:);
popmax=min(popmax1,input(2,2:15)+40*delta);
popmin=max(popmin1,input(2,2:15)-40*delta);
for MMM=1:321
    % 产生初始粒子和速度
for i=1:sizepop
    %随机产生一个种群
    pop(i,1)=input(MMM,1);       %初始种群
    V(i,1)=0;    %初始化速度
    for j=2:15

pop(i,j)=input(MMM,j);%popmin(j-1)+(popmax(j-1)-popmin(j-1))*rand(1,1);       %初始种群
            if rand(1,1)>0.5
                V(i,j)=delta(1,j-1);    %初始化速度
            else
                V(i,j)=-delta(1,j-1);
            end
        end

        %计算适应度
```

```
        fitness(i)=fun(pop(i,:),MMM);     %染色体的适应度
    end

    %% 个体极值和群体极值
    [bestfitness, bestindex]=min(fitness);
    zbest(MMM,:)=pop(bestindex,:);     %全局最佳
    gbest=pop;       %个体最佳
    fitnessgbest=fitness;     %个体最佳适应度值
    fitnesszbest=bestfitness;     %全局最佳适应度值

    %% 迭代寻优
    for i=1:maxgen
        for j=1:sizepop
            for k=2:15
                %速度更新
                if rand(1,1)>0.5
                    V(j,k)=delta(1,k-1);   %初始化速度
                else
                    V(j,k)=-delta(1,k-1);
                end
                %V(j,k) = V(j,k) + c1*rand*(gbest(j,k) - pop(j,k)) + c2*rand*(zbest(1,k) -
pop(j,k));
                %V(j,find(V(j,k)>Vmax))=Vmax;
                %V(j,find(V(j,k)<Vmin))=Vmin;
                V(j,1)=0;   %初始化速度

                %种群更新
                if i>1&&fitness(1,j)>130
                    pop(j,k)=pop(j,k);
                else
                    pop(j,k)=pop(j,k)+1.0*V(j,k);
                    pop(j,find(pop(j,k)>popmax(k-1)))=popmax(k-1);
                    pop(j,find(pop(j,k)<popmin(k-1)))=popmin(k-1);
                    pop(j,1)=input(MMM,1);       %初始种群
                end

            end


            %if rand>0.98
            %       for j=2:17
            %            pop(i,j)=popmin(j-1)+(popmax(j-1)-popmin(j-1))*rand(1,1);       %初
始种群
            %       end
            %end

            %适应度值
            fitness(j)=fun(pop(j,:),MMM);
```

```
    end

    for j=1:sizepop

        %个体最优更新
        if fitness(j) < fitnessgbest(j)
            gbest(j,:) = pop(j,:);
            fitnessgbest(j) = fitness(j);
        end

        %群体最优更新
        if fitness(j) < fitnesszbest
            zbest(MMM,:) = pop(j,:);
            fitnesszbest = fitness(j);
        end
    end
    yy(i)=fitnesszbest;

end

end
%% 结果分析
plot(yy)
title('最优个体适应度','fontsize',12);
xlabel('进化代数','fontsize',12);ylabel('适应度','fontsize',12);
```

问题五主要程序

```
%% 该代码为基于变异粒子群算法的函数极值寻优算法
%% 参数初始化
%粒子群算法中的两个参数
%c1 = 1.19445;
%c2 = 1.19445;

maxgen=40;     % 进化次数
sizepop=70;    %种群规模

%Vmax=1;
%Vmin=-1;
load data input
load data1 net1 inputps outputps1
mm=xlsread('maxmin.xlsx');
popmax1=mm(2,:);
popmin1=mm(1,:);
delta=mm(3,:);
popmax=min(popmax1,input(2,2:15)+40*delta);
popmin=max(popmin1,input(2,2:15)-40*delta);
```

```
        % 产生初始粒子和速度
    for i=1:sizepop
        %随机产生一个种群
        pop(i,1)=input(133,1);       %初始种群
        V(i,1)=0;   %初始化速度
        for j=2:15
            pop(i,j)=input(133,j);%popmin(j-1)+(popmax(j-1)-popmin(j-1))*rand(1,1);       %
初始种群
            if rand(1,1)>0.5
                V(i,j)=delta(1,j-1);   %初始化速度
            else
                V(i,j)=-delta(1,j-1);
            end
        end

        %计算适应度
        fitness(i)=fun(pop(i,:));     %染色体的适应度
    end
    save 0 pop V

    %% 个体极值和群体极值
    [bestfitness, bestindex]=min(fitness);
    zbest=pop(bestindex,:);     %全局最佳
    gbest=pop;       %个体最佳
    fitnessgbest=fitness;     %个体最佳适应度值
    fitnesszbest=bestfitness;     %全局最佳适应度值

    %% 迭代寻优
    for i=1:maxgen
        for j=1:sizepop


            for k=2:15
                %速度更新
                if rand(1,1)>0.5
                    V(j,k)=delta(1,k-1);   %初始化速度
                else
                    V(j,k)=-delta(1,k-1);
                end
                %V(j,k) = V(j,k) + c1*rand*(gbest(j,k) - pop(j,k)) + c2*rand*(zbest(1,k) -
pop(j,k));
                %V(j,find(V(j,k)>Vmax))=Vmax;
                %V(j,find(V(j,k)<Vmin))=Vmin;
                V(j,1)=0;   %初始化速度

                %种群更新
                if i>1&&fitness(1,j)>130
                    pop(j,k)=pop(j,k);
```

```
        else
            pop(j,k)=pop(j,k)+1.0*V(j,k);
            pop(j,find(pop(j,k)>popmax(k-1)))=popmax(k-1);
            pop(j,find(pop(j,k)<popmin(k-1)))=popmin(k-1);
            pop(j,1)=input(133,1);       %初始种群
        end

    end


    %if rand>0.98
    %       for j=2:17
    %           pop(i,j)=popmin(j-1)+(popmax(j-1)-popmin(j-1))*rand(1,1);       %初
始种群
    %       end
    %end

    %适应度值
    if i==1
        save 1 pop V
    elseif i==2
        save 2 pop V
    elseif i==3
        save 3 pop V
    elseif i==4
        save 4 pop V
    elseif i==5
        save 5 pop V
    elseif i==6
        save 6 pop V
    elseif i==7
        save 7 pop V
    elseif i==8
        save 8 pop V
    elseif i==9
        save 9 pop V
    elseif i==10
        save 10 pop V
    elseif i==11
        save 11 pop V
    elseif i==12
        save 12 pop V
    elseif i==13
        save 13 pop V
    elseif i==14
        save 14 pop V
    elseif i==15
        save 15 pop V
    elseif i==16
```

```
        save 16 pop V
elseif i==17
        save 17 pop V
elseif i==18
        save 18 pop V
elseif i==19
        save 19 pop V
elseif i==20
        save 20 pop V
elseif i==21
        save 21 pop V
elseif i==22
        save 22 pop V
elseif i==23
        save 23 pop V
elseif i==24
        save 24 pop V
elseif i==25
        save 25 pop V
elseif i==26
        save 26 pop V
elseif i==27
        save 27 pop V
elseif i==28
        save 28 pop V
elseif i==29
        save 29 pop V
elseif i==30
        save 30 pop V
elseif i==31
        save 31 pop V
elseif i==32
        save 32 pop V
elseif i==33
        save 33 pop V
elseif i==34
        save 34 pop V
elseif i==35
        save 35 pop V
elseif i==36
        save 36 pop V
elseif i==37
        save 37 pop V
elseif i==38
        save 38 pop V
elseif i==39
        save 39 pop V
elseif i==40
        save 40 pop V
end
```

```
        fitness(j)=fun(pop(j,:));

    end

    for j=1:sizepop

        %个体最优更新
        if fitness(j) < fitnessgbest(j)
            gbest(j,:) = pop(j,:);
            fitnessgbest(j) = fitness(j);
        end

        %群体最优更新
        if fitness(j) < fitnesszbest
            M=i;
            zbest = pop(j,:);
            save Zbest zbest
            fitnesszbest = fitness(j);
        end
    end
    yy(i)=fitnesszbest;

end

%% 结果分析
plot(yy)
title('最优个体适应度','fontsize',12);
xlabel('进化代数','fontsize',12);ylabel('适应度','fontsize',12);
```